Statistica Sinica **34** (2024), 1391-1412
doi:https://doi.org/10.5705/ss.202021.0393

NONLINEAR DIMENSION REDUCTION FOR FUNCTIONAL DATA WITH APPLICATION TO CLUSTERING

Ruoxu Tan*, Yiming Zang and Guosheng Yin

Tongji University, Université de Lorraine and Imperial College London

Abstract: Functional data often possess nonlinear structures, for example, phase variation, for which linear dimension-reduction techniques can be ineffective. We study nonlinear dimension reduction for functional data based on the assumption that the data lie on an unknown manifold contaminated with noise. We generalize a recently developed manifold learning method designed for high-dimensional data into our context, and derive asymptotic convergence results, taking noise into account. The results based on synthetic examples often produce more accurate geodesic distance estimations than those of the traditional functional Isomap method. We further develop a clustering strategy based on the manifold learning outcomes, and demonstrate that our method outperforms others if the data lie on a curved manifold. Two real-data examples are presented for illustration.

Key words and phrases: Geodesic distance, graph clustering, manifold learning, measurement error.

1. Introduction

Popular methods of dealing with high-/infinite-dimensional data reduce the dimension of the data, for example, using a principal component analysis or a linear discriminant analysis. More recently, nonlinear methods such as manifold learning have been developed to handle complex data patterns, particularly for high-dimensional data. Well-known methods include Isomap (Tenenbaum, De Silva and Langford. (2000)), local linear embedding (Roweis and Saul (2000)), Laplacian eigenmaps (Belkin and Niyogi (2003)), tangent space alignment (Zhang and Zha (2004)), and vector diffusion maps (Singer and Wu (2012)). These methods and their variants have been used successfully in many fields, for example, in imaging data analysis (Pless and Souvenir (2009)) when the pixels lie in a high-dimensional vector space, but are concentrated on a low-dimensional manifold.

Functional data are usually collected sequentially over time. Unlike high-dimensional data, functional data are intrinsically infinite dimensional, and thus the demand for dimension reduction is more pressing. Classical functional principal component analysis (FPCA) is a core technique of linear dimension

*Corresponding author.

reduction in functional data analysis. However, it may need a large number of components to explain an intrinsically low-dimensional data set (Lin and Yao (2020)), and the nonlinear structures might still not be explained adequately. For example, an FPCA often produces "horseshoe"-shape plots of principal components in the presence of phase variation, which is ubiquitous in functional data (Chen and Müller (2012)). Phase variation is a major cause of nonlinearity in functional data. It can be summarized as a common shape feature presented in different parts of the domain across individuals; for example, see the various peaks of the Berkeley growth velocity data in Figure 1. The sample mean curve possesses less significant shape features and the variance is enlarged by the phase variation. These nonlinear features make an FPCA an ineffective approach. To "unwrap" this nonlinearity of phase variation, manifold techniques have been proposed (Kneip and Gasser (1992); Srivastava et al. (2011); Chen and Müller (2012)). However, few studies assume an unknown manifold structure for functional data: Chen and Müller (2012) propose nonlinear variants of the FPCA, and Lin and Yao (2020) examine functional regression. In contrast to manifold-domain or manifold-valued problems, in which the manifold is known a priori (e.g., Lila, Aston and Sangalli (2016); Dai and Müller (2018); Lin and Yao (2019); Dai, Lin and Müller (2020); Lin, Shao and Yao (2020); Zhang and Saparbayeva (2021)), we focus on manifold learning for functional data with an unknown manifold.

Apart from nonlinear structures, a notable difficulty in a functional data analysis is that the data are rarely recorded continuously, but only discretely, with errors. Therefore, a function does not lie exactly on the manifold, even after smoothing, for any finite sample. Chen and Müller (2012) propose the penalized functional Isomap to mitigate this problem. Here, we investigate tangent spaces and parallel transport in functional manifold learning, and propose functional parallel transport unfolding (FPTU) to produce more robust geodesic distance estimates. This idea was proposed in Budninskiy et al. (2019) for an error-free high-dimensional data setting. We further develop the asymptotic consistency of the geodesic distance estimation of FPTU by taking noise into account. Using synthetic examples, we show that FPTU often produces more accurate geodesic distance estimates than those of a functional Isomap.

As an application of the functional manifold learning procedures, we propose a new clustering strategy for functional data based on the manifold learning outcomes. Here, we use the geodesic distance instead of the L^2 (Euclidean) distance to quantify the proximity within data. We show using several classical synthetic examples that our new strategy outperforms other methods that do not take the manifold structure into account. We further apply our method to two real-data examples, namely, Berkeley growth data and yeast gene expression data, and demonstrate that new insights can be gained based on the proposed functional manifold learning techniques.

The remainder of this paper is structured as follows. In Section 2, we present the model and data, followed by the classical functional Isomap and our proposed FPTU. Here, we also include guidelines on selecting the tuning parameters. In Section 3, we establish the asymptotic consistency of the geodesic distance estimation of FPTU. A new clustering strategy based on the manifold learning outcomes is introduced in Section 4. Sections 5 and 6 show simulations and real-data examples, respectively. We conclude the paper in Section 7. Technical proofs are deferred to the Supplementary Material.

2. Nonlinear Dimension Reduction

2.1. Model and data

Let X be a real-valued second-order random process defined on a compact interval $[0, 1]$, without loss of generality. Although X naturally belongs to the ambient space $L^2([0, 1])$, we assume that X lies on a low-dimensional unknown functional manifold $\mathcal{M}$ of intrinsic dimension d, where $\mathcal{M}$ is assumed to be an embedded manifold in $L^2([0, 1])$, with the metric induced from the L^2 metric. More precisely, let $\mathcal{M}_{\text{st}}$ be a d-dimensional differentiable manifold in the usual sense, and let $\iota : \mathcal{M}_{\text{st}} \to L^2([0, 1])$ be an embedding, that is, a diffeomorphism onto its image. Then, $\iota(\mathcal{M}_{\text{st}}) \equiv \mathcal{M} \subset L^2([0, 1])$ is the functional manifold that we are interested in, and X lies on a manifold such that for all ω in the sample space Ω, $X(\omega) \in \mathcal{M}$. Note that $\mathcal{M}$ is a d-dimensional functional subspace for which we have little prior information. This differs from functional data valued in a known submanifold of a vector space, for example, as in Dai and Müller (2018) and Lin and Yao (2019). Because the manifold $\mathcal{M}$ is, in general, curved, the L^2 distance $\|\cdot\|_{L^2}$ is not a proper measure to quantify the proximity of the elements on $\mathcal{M}$. A more appropriate choice is the geodesic distance $d_g(\cdot, \cdot)$, taking the intrinsic structure of $\mathcal{M}$ into account. Similarly to classical manifold learning, our goal is to estimate geodesic distances so that we can represent functional data using low-dimensional coordinates based on these distances.

Let $\{X_i\}_{i=1}^n$ be an independent and identically distributed (i.i.d.) sample of X. In practice, functional data are rarely recorded continuously. We often only observe discrete and noisy data $(T_{i,j}, Y_{i,j})$, satisfying the model

$$Y_{i,j} = X_i(T_{i,j}) + \epsilon_{i,j}\,, \quad i = 1, \ldots, n;\ j = 1, \ldots, J_i, \tag{2.1}$$

where $T_{i,j} \in [0, 1]$ are the random time points of observing $Y_{i,j}$, and $\epsilon_{i,j}$ are i.i.d. mean zero random errors with $\text{Var}(\epsilon_{i,j}) = \sigma^2 < \infty$. Our first task is to recover the functions X_i from the discrete data $(T_{i,j}, Y_{i,j})$. If the sampling is not too sparse, that is, $\inf_i J_i$ is sufficiently large, we can estimate X_i using individual smoothing, for example, by applying a local polynomial estimation (Fan and Gijbels (1996)) on each individual data $\{(T_{i,1}, Y_{i,1}), \ldots, (T_{i,J_i}, Y_{i,J_i})\}$,

for $i = 1, \ldots, n$. If the sampleing is too sparse, so that individual smoothing cannot produce reasonable estimates, other techniques (e.g., Yao, Müller and Wang (2005)) may be applied. Assuming that the sampling is not too sparse, we use the ridged local linear estimator (Lin and Yao (2020)) to obtain the estimator $\hat{X}_i$ of X_i. Specifically, a standard local linear estimator $\tilde{X}_i(t) = \tilde{a}_0$ can be obtained by minimizing a sum of weighted least squares,

$$(\tilde{a}_0, \tilde{a}_1) = \underset{a_0, a_1}{\operatorname{argmin}} \sum_{j=1}^{J_i} \{Y_{i,j} - a_0 - a_1(T_{i,j} - t)\}^2 \mathcal{K}_{h_i}(T_{i,j} - t)\,,$$

where $\mathcal{K}_h(\cdot) = \mathcal{K}(\cdot/h)/h$, $\mathcal{K}$ is a kernel function, and $h_i > 0$ is a bandwidth. In practical computing, $\mathcal{K}$ is often a symmetric density function, and h can be chosen using any standard procedure, such as cross-validation or plug-in methods (Fan and Gijbels (1996)). The closed-form solution is given by $\tilde{X}_i(t) = (T_0 S_2 - T_1 S_1)/(S_0 S_2 - S_1^2)$, where

$$S_r = \frac{1}{J_i} \sum_{j=1}^{J_i} \mathcal{K}_{h_i}(T_{i,j} - t) \left(\frac{T_{i,j} - t}{h_i} \right)^r, \; T_r = \frac{1}{J_i} \sum_{j=1}^{J_i} \mathcal{K}_{h_i}(T_{i,j} - t) \left(\frac{T_{i,j} - t}{h_i} \right)^r Y_{i,j}\,,$$

for $r = 0, 1$, and 2. To obtain more stable estimates when the denominator is close to zero, we introduce a ridge parameter $\lambda > 0$,

$$\hat{X}_i(t) = \frac{T_0 S_2 - T_1 S_1}{S_0 S_2 - S_1^2 + \lambda \operatorname{sign}(S_0 S_2 - S_1^2) \mathbb{1}\{|S_0 S_2 - S_1^2| < \lambda\}}\,, \tag{2.2}$$

where λ can be set to J_i^{-2}, following Lin and Yao (2020). The $\hat{X}_i$ are subject to estimation errors, and so do not lie exactly on the manifold $\mathcal{M}$, which poses difficulties from both practical and theoretical aspects.

2.2. Functional Isomap

In the classical high-dimensional data space, if two data points are sufficiently close, then their Euclidean distance is a good approximation of their geodesic distance. This is also true for functional data if the Euclidean distance is replaced with the L^2 distance, and $\mathcal{M}$ is an embedded manifold in $L^2([0,1])$, with the metric induced by the L^2 metric. Therefore, we use the L^2 distance to approximate the local geodesic distance. Then, the estimated geodesic between two far-away functions is defined as the shortest path moving along the line segments (local geodesics), and the geodesic distance is estimated as the sum of the L^2 distances of the line segments along the path.

Specifically, for each $\hat{X}_i$, its local neighborhood can be defined by an ϵ-ball or K-nearest neighbors (K-NN). To better control the number of individuals in a neighborhood, we define the neighborhood $\mathcal{N}_i$ of $\hat{X}_i$ as the K individuals closest to $\hat{X}_i$, based on the L^2 distance. We then construct the proximity graph G, as

follows. First, the vertices of G include all data points. Second, two points i and j are linked if $i \in \mathcal{N}_j$ and $j \in \mathcal{N}_i$, and the edge is weighted by the L^2 distance $\|\hat{X}_i - \hat{X}_j\|_{L^2} = [\int_0^1 \{\hat{X}_i(t) - \hat{X}_j(t)\}^2 \, dt]^{1/2}$. Subsequently, the geodesic path between any two points can be found by applying Dijkstra's algorithm on G. The geodesic path from i to j is an ordered sequence of indices representing the shortest path from i to j on G, which is a discrete approximation of the true geodesic. The geodesic distance between i and j is estimated using the length of the geodesic path. It may happen that G is not connected, owing to a small K in K-NN or the presence of outliers, in which case, we consider each connecting component separately. To simplify the presentation, we assume throughout Section 2 that G is connected. The final step is to apply multi-dimensional scaling (MDS; Cox and Cox (2008)) on G to obtain low-dimensional coordinates of our data.

The Isomap idea is also adopted in the functional data setting by Chen and Müller (2012). However, estimation errors from discrete and noisy observations $(T_{i,j}, Y_{i,j})$ may make the resulting geodesic distances unstable. Chen and Müller (2012) proposed penalizing low-density areas in the search of geodesic paths, while still using the sum of the L^2 distances of the resulting path as the estimated geodesic distance. However, this does not adjust for estimation errors in functional data. As an alternative, we propose a procedure that produces more robust estimations of geodesic distances. Our method can be viewed as the functional version of the method proposed by Budninskiy et al. (2019).

2.3. Functional parallel transport unfolding

We use G defined in Section 2.2 to obtain the geodesic paths. However, we estimate the geodesic distance using parallel transport unfolding, instead of the sum of the L^2 distances of the line segments as in Isomap. Parallel transport provides a way of moving tangent vectors between different tangent spaces without losing geometric information, such as angles and lengths. Because the manifold $\mathcal{M}$ is unknown, we compute a *discrete* parallel transport using a functional version of tangent space alignment, developed by Singer and Wu (2012) in the high-dimensional data setting.

The standard method for estimating the tangent space at X_i, $T_i\mathcal{M} \equiv T_{X_i}\mathcal{M}$, is based on a local (functional) principal component analysis (PCA, Singer and Wu (2012); Lin and Yao (2020)). Recall that d is the intrinsic dimension of $\mathcal{M}$. We assume d to be known, for the time being. In Section 2.4, we introduce a method for estimating d when it is unknown. The tangent space $T_i\mathcal{M}$ is the best linear approximation of $\mathcal{M}$ locally around X_i. To estimate such a linear space, we use the space spanned by the first d eigenfunctions of the estimated local covariance function of X, computed from the neighborhood of $\hat{X}_i$ (because the true X_i is not available). Specifically, let $\mathcal{N}_{i,\mathrm{PCA}}$ be the neighborhood of $\hat{X}_i$ containing K_{PCA}-NN of $\hat{X}_i$ (see Section 2.4 for suggestions on how to choose

K_{PCA}). We define the local empirical covariance function around $\hat{X}_i$ as

$$\hat{\Gamma}_i(s,t) = \frac{1}{K_{\mathrm{PCA}}} \sum_{j \in \mathcal{N}_{i,\mathrm{PCA}}} \{\hat{X}_j(s) - \hat{\mu}_i(s)\}\{\hat{X}_j(t) - \hat{\mu}_i(t)\}, \quad \text{for } s,t \in [0,1],$$

where $\hat{\mu}_i = \sum_{j \in \mathcal{N}_{i,\mathrm{PCA}}} \hat{X}_j / K_{\mathrm{PCA}}$.

Remark 1. In the neighborhood of $\hat{X}_i$, the data are centered using $\hat{\mu}_i$ instead of $\hat{X}_i$. We found that this slightly improves the numerical performance of our estimator, possibly because $\hat{\mu}_i$ averages out the estimation errors of functional data. It is not difficult to show that the two choices have the same asymptotic properties.

The first d eigenfunctions corresponding to the largest d eigenvalues of $\hat{\Gamma}_i$ are denoted by $\hat{\Phi}_i = (\hat{\phi}_{i1}, \ldots, \hat{\phi}_{id})$, and we define the estimated tangent space as $\hat{T}_i\mathcal{M} = \mathrm{span}\{\hat{\phi}_{i1}, \ldots, \hat{\phi}_{id}\}$. The next step is to construct parallel transport between the tangent spaces. Suppose $\hat{X}_i \in \mathcal{N}_j$, and we aim to parallel transport vectors from $\hat{T}_i\mathcal{M}$ to $\hat{T}_j\mathcal{M}$. If $\hat{T}_i\mathcal{M}$ and $\hat{T}_j\mathcal{M}$ were the same space, then $\hat{\Phi}_i$ and $\hat{\Phi}_j$ would differ only by an orthogonal transformation; that is, there exists an orthogonal matrix $R_{j,i} \in \mathcal{O}(d)$ such that $\hat{\Phi}_i = \hat{\Phi}_j R_{j,i}$, where $\mathcal{O}(d)$ denotes the group of $d \times d$ orthogonal matrices. However, $\hat{T}_i\mathcal{M}$ and $\hat{T}_j\mathcal{M}$ are typically different. Following Singer and Wu (2012), we define $\hat{R}_{j,i}$ as the orthogonal matrix that minimizes the Frobenius norm (also called the Hilbert–Schmidt norm in functional analysis) of $\hat{\Phi}_i - \hat{\Phi}_j R$, for $R \in \mathcal{O}(d)$,

$$\hat{R}_{j,i} = \operatorname*{argmin}_{R \in \mathcal{O}(d)} \|\hat{\Phi}_i - \hat{\Phi}_j R\|_{\mathrm{F}}^2 = \operatorname*{argmin}_{R \in \mathcal{O}(d)} \sum_{k=1}^{d} \int_0^1 \left\{ \hat{\phi}_{ik}(t) - \sum_{s=1}^{d} r_{s,k} \hat{\phi}_{js}(t) \right\}^2 dt, \quad (2.3)$$

where $r_{s,k}$ denotes the (s,k)th entry of R. There is a closed-form solution for $\hat{R}_{j,i}$, as indicated by the following result.

Proposition 1. *Let $[\hat{\Phi}_i^T, \hat{\Phi}_j]$ be a $d \times d$ matrix with the (k,s)th entry $\langle \hat{\phi}_{ik}, \hat{\phi}_{js} \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the inner product of $L^2[0,1]$. Let $U\Sigma V^T$ be the singular value decomposition of $[\hat{\Phi}_i^T, \hat{\Phi}_j]$, and then we have*

$$\hat{R}_{j,i} = VU^T. \quad (2.4)$$

The proof is given in Supplement A of the Supplementary Material, and it can be shown that $\hat{R}_{j,i}$ provides an approximation of the parallel transport from $\hat{T}_i\mathcal{M}$ to $\hat{T}_j\mathcal{M}$. That is, for a vector $u \in \hat{T}_i\mathcal{M}$ with coordinates $(u_1, \ldots, u_d)^T$ in terms of the basis $\hat{\Phi}_i$, $\hat{R}_{j,i}(u_1, \ldots, u_d)^T$ gives the approximate coordinates of the parallel transported $u \in \hat{T}_j\mathcal{M}$ under the basis $\hat{\Phi}_j$.

Suppose that $(i = i_0, i_1, \ldots, i_{m-1}, i_m = j)$ is a geodesic path in G. We iteratively parallel transport each of the edges of the path to the last tangent

space $\hat{T}_j\mathcal{M}$ and then aggregate them. This results in an "unfolded geodesic" (straight line) in $\hat{T}_j\mathcal{M}$, the norm of which provides a good approximation of the geodesic distance. Specifically, we first project $\hat{V}_{i_k} = \hat{X}_{i_{k-1}} - \hat{X}_{i_k}$ to the tangent space $\hat{T}_{i_k}\mathcal{M}$, for $k = 1, \ldots, m$. Let $v_{i_k} = (\langle \hat{V}_{i_k}, \hat{\phi}_{i_k 1}\rangle, \ldots, \langle \hat{V}_{i_k}, \hat{\phi}_{i_k d}\rangle)^T$ be the coordinates of the projected $\hat{V}_{i_k}$. Next, they are parallel transported to $\hat{T}_j M$ with coordinates

$$v_{i_k,m} = \left(\prod_{s=k}^{m-1} \hat{R}_{i_{s+1},i_s} \right) v_{i_k}\,, \text{ for } k = 1, \ldots, m-1\,.$$

For $k = m$, $v_{i_k} = v_j$ is already in $\hat{T}_j\mathcal{M}$, and thus no parallel transport is needed, and we define $v_{j,m} = v_j$. Finally, we use the aggregated coordinates $v_{i,m} = \sum_{k=1}^{m} v_{i_k,m}$ to denote the unfolded geodesic from $\hat{X}_i$ to $\hat{X}_j$ in $\hat{T}_j\mathcal{M}$. The Euclidean norm of $v_{i,m}$, or equivalently, the L^2 norm of $\hat{\Phi}_j v_{i,m}$, is a robust approximation of the geodesic distance.

Remark 2. As suggested in Budninskiy et al. (2019), projecting $\hat{V}_{i_k}$ to $\hat{T}_{i_k}\mathcal{M}$ discards the tail information of $\hat{V}_{i_k}$ that is possibly caused by noise. This may be beneficial in the case of large noise. On the other hand, in the case of small or no noise, the L^2 norm of $\hat{V}_{i_k}$ is a sufficiently good approximation of the local geodesic distance. In this case, we may use the L^2 norm of $\hat{V}_{i_k}$ directly by rescaling v_{i_k}: that is, to replace v_{i_k} with $v'_{i_k} = v_{i_k}\|\hat{V}_{i_k}\|_{L^2}/\|v_{i_k}\|$, where $\|\cdot\|$ denotes the Euclidean norm.

The main advantage of FPTU is that it considers the geodesic curvature. It tries to preserve the intrinsic angles between v_{i_k}. Thus, the unrolled polyline is approximately a straight line in $\mathbb{R}^d$, provided that the geodesic path in G is sufficiently close to the true geodesic on $\mathcal{M}$. However, in general, owing to finite sampling and estimation errors in functional data, the geodesic path in G is twisted, and thus the unrolled polyline is not straight. FPTU uses an aggregated vector (straight line) that ignores the twists and turns of the path to mitigate such errors from finite samples. In contrast, functional Isomap does not consider the intrinsic angles and uses only the L^2 lengths of the line segments, which is sensitive to irregular sampling and estimation errors. In particular, as pointed out by Budninskiy et al. (2019), Isomap introduces distortions if the sampling domain is not geodesically convex, whereas PTU is able to handle such a situation.

Because the resulting distance from $\hat{X}_i$ to $\hat{X}_j$ is, in general, different to the one from $\hat{X}_j$ to $\hat{X}_i$, we define our final estimate of the geodesic distance $\hat{d}_g(\hat{X}_i, \hat{X}_j)$ as the average of these two. Once we have the geodesic distances between any two points, we apply the classical MDS to obtain the d-dimensional representations, denoted by $\hat{Z}_i$, of the data.

2.4. Selection of tuning parameters

One tuning parameter in our procedure is K, which is used to define the proximity graph G, and another is K_{PCA}, which is used to estimate tangent spaces. As suggested in Budninskiy et al. (2019), using $K_{\mathrm{PCA}} = K$ often yields good results, as shown in our numerical examples. In theory, $K_{\mathrm{PCA}} \asymp n^{2/(d+2)}$ is a consistent choice (see Condition (B2) in Section 3). Therefore, we suggest experimenting with several values around $n^{2/(d+2)}$, for both K_{PCA} and K. The value of K_{PCA} has a smaller effect on the performance of FPTU, once K is fixed. In the case of very noisy observations, a larger K_{PCA} may be beneficial.

In practice, the intrinsic dimension d is often unknown, especially for functional data. Facco et al. (2017) developed an estimator of d using minimal neighborhood information for high-dimensional data. To adapt their method to the functional data setting, we first use the standard (global) functional PCA to reduce the $\hat{X}_i$ to the PC scores $\xi_i = (\xi_{i1}, \ldots, \xi_{i\ell})$, where ℓ can be chosen using any standard method, say, explaining 95% of the variance (Ramsay and Silverman (2005)). This step also mitigates the effect of noise. We then compute the pairwise Euclidean distances between the ξ_i. For each individual i, let $\rho_i = r_{i2}/r_{i1}$, where r_{i1} and r_{i2} are the shortest and the second shortest distances, respectively, to ξ_i. Next, we sort the ρ_i in ascending order $\{\rho_{(1)}, \ldots, \rho_{(n)}\}$, and define the empirical distribution function $F(\rho_{(i)}) = i/n$, for $i = 1, \ldots, n$. Facco et al. (2017) showed that $-\log\{1 - F(\rho)\}/\log(\rho) = d$, for $\rho \in [1, \infty)$. Therefore, we estimate d using the integer closest to $\operatorname{argmin} \sum_{i \in N_\delta} [-\log\{1 - F(\rho_i)\} - d \log(\rho_i)]^2$, where δ is a given value in $(0, 1)$ and N_δ is the index set corresponding to $\{\rho_{(1)}, \ldots, \rho_{([n\delta])}\}$ ($[\cdot]$ denotes rounding to integer). That is, the largest $n - [n\delta]$ values of ρ_i are not used, because, as pointed out by Facco et al. (2017), larger values of ρ_i are often caused by irregular sampling and can significantly destabilize the final estimate. Note that at least the largest ρ_i should be discarded, because $F(\rho_{(n)}) = 1$, and thus $-\log\{1 - F(\rho_{(n)})\} = \infty$. We use $\delta = 0.9$ in our numerical study; a slightly smaller δ often yields the same estimate of d.

3. Theoretical Properties

To establish the asymptotic consistency of the geodesic distance estimation of FPTU, we generalize the theory in the finite-dimensional setting to the functional setting, while taking the estimation errors of functional data into account. We assume that both $T_{i,j}$ and $\epsilon_{i,j}$ in (2.1) are i.i.d. and that $\epsilon_{i,j}$ is independent of $T_{i,j}$. The subscript is omitted for the notation of a generic variable. We focus on the case in which the sampling on each individual is sufficiently dense, that is, $J_i \asymp J$, for all i and $J \to \infty$. Recall that we use the ridged local linear estimator in (2.2) to estimate X. Let h be the bandwidth used in (2.2) corresponding to J. The following conditions are needed to ensure the consistency of $\hat{X}$:

Condition A.

(A1) X is twice continuously differentiable, with $\|X''\|_\infty = O_p(1)$.

(A2) The sampling density f_T is twice continuously differentiable, with $\|f_T''\|_\infty < \infty$ and $\inf_{t\in[0,1]} f_T(t) > 0$.

(A3) Compactly supported on $[-1,1]$, the kernel $\mathcal{K}$ is differentiable with a bounded derivative, and satisfies $\int_{-1}^{1} \mathcal{K}(u)\,du = 1$, $\int_{-1}^{1} u\mathcal{K}(u)\,du = 0$, and $\int_{-1}^{1} u^2\mathcal{K}(u)\,du < \infty$.

(A4) $J \to \infty$, $h \asymp J^{-1/5}$, and $\lambda = O(J^{-2})$ as the sample size $n \to \infty$.

The above conditions are mild and similar to those used in the literature on local polynomial smoothing; see Fan and Gijbels (1996) and Lin and Yao (2020). In Condition (A4), $h \asymp J^{-1/5}$ is optimal in terms of the mean squared error, and λ is chosen to be of order $O(J^{-2})$ so that the ridge term is asymptotically negligible.

To list the conditions for the manifold structure, first recall that $\mathcal{M}$ is an embedded manifold of $L^2([0,1])$, with the metric induced by the L^2 metric. We introduce h_{PCA} as the maximum radius of K_{PCA}-NNs used to estimate $T_i\mathcal{M}$, so that $h_{\mathrm{PCA}} = O\{(K_{\mathrm{PCA}}/n)^{1/d}\}$. Note that K_{PCA}/n approximates the probability that X falls within the region of K_{PCA}-NN, which is approximately proportional to h_{PCA}^d. Let $\mathcal{P}_{j,i}$: $T_i\mathcal{M} \to T_j\mathcal{M}$ be the parallel transport operator from $T_i\mathcal{M}$ to $T_j\mathcal{M}$, and let $\tilde{\Phi}_i = (\tilde{\phi}_{i1}, \ldots, \tilde{\phi}_{id})$ be the version of $\hat{\Phi}_i = (\hat{\phi}_{i1}, \ldots, \hat{\phi}_{id})$ based on the true X_i.

Condition B

(B1) The probability density f_X of X on $\mathcal{M}$ satisfies $0 < c_1 < \inf_{x\in\mathcal{M}} f_X(x) < \sup_{x\in\mathcal{M}} f_X(x) < c_2 < \infty$, for some constants c_1 and c_2.

(B2) $h_{\mathrm{PCA}} \to 0$ and $h_{\mathrm{PCA}} \gtrsim \max\{J^{-2/5+\epsilon}, n^{-1/(d+2)}\}$, for a small but fixed $\epsilon > 0$, as $n \to \infty$.

(B3) For all i and j, there exists $\varepsilon > 0$ such that the smallest singular value of $[\Phi_i^T, \Phi_j]$ is larger than ε.

(B4) For any geodesic path $(i = i_0, i_1, \ldots, i_{m-1}, i_m = j)$ in G, assume that the polyline $(X_{i_0}, \ldots, X_{i_m})$ is included in an h_d-thickening of the true geodesic between X_{i_0} and X_{i_m}, where h_d is a positive constant such that $h_d\kappa_s \ll 1$, with κ_s being the maximum absolute value of the intrinsic sectional curvature of $\mathcal{M}$.

(B5) For any two points X_k and X_ℓ on $\mathcal{M}$ such that the geodesic distance between them is $O(h_g)$, for some $h_g > 0$, let $u_{k\ell}$ be the tangent vector in

$T_\ell\mathcal{M}$ that connects the endpoints of the true geodesic curve between X_k and X_ℓ mapped onto $T_\ell\mathcal{M}$, and assume

$$(\langle u_{k\ell}, \tilde{\phi}_{\ell 1}\rangle, \ldots, \langle u_{k\ell}, \tilde{\phi}_{\ell d}\rangle) = (\langle X_k - X_\ell, \tilde{\phi}_{\ell 1}\rangle, \ldots, \langle X_k - X_\ell, \tilde{\phi}_{\ell d}\rangle) + O_p(h_{\mathrm{PCA}} + h_g^3)\,.$$

Condition (B1) is mild and ensures regular sampling. In Condition (B2), $h_{\mathrm{PCA}} \asymp n^{-1/(d+2)}$ is the standard order used in Singer and Wu (2012) and Budninskiy et al. (2019) to ensure the consistency of tangent space estimations. Note that $h_{\mathrm{PCA}} \gtrsim J^{-2/5+\epsilon}$ is introduced so that the optimal convergence rate of $E(\|\hat{X} - X\|_{L^2}^2 | X)^{1/2}$, $J^{-2/5}$, decays faster than h_{PCA}. It is introduced following Lin and Yao (2020) to simplify the exposition of the theoretical results, but it does not impose any additional restriction. Condition (B3) ensures the uniqueness of the orthogonal matrix for the discrete parallel transport. Conditions (B4) and (B5) are adopted from Budninskiy et al. (2019), and are related to the regularity of the manifold and sampling.

Theorem 1. *Under Conditions* (A1)–(A4) *and* (B1)–(B5)*:*

(I) The discrete parallel transport defined in Section 2.3 *converges in probability to the true parallel transport in the sense that, for any two points X_k and X_ℓ on $\mathcal{M}$ such that $d_g(X_k, X_\ell) = O(h_g)$ with some $h_g > 0$, and for $u_k \in \hat{T}_k\mathcal{M}$ and $u_{k0} \in T_k\mathcal{M}$ such that $\|u_k - u_{k0}\|_{L^2} = O(h_{\mathrm{PCA}})$, we have*

$$\begin{aligned}&\hat{R}_{\ell,k}(\langle u_k, \hat{\phi}_{k1}\rangle, \ldots, \langle u_k, \hat{\phi}_{kd}\rangle)^T \\ &= (\langle \mathcal{P}_{\ell,k}(u_{k0}), \tilde{\phi}_{\ell 1}\rangle, \ldots, \langle \mathcal{P}_{\ell,k}(u_{k0}), \tilde{\phi}_{\ell d}\rangle)^T + O_p(h_{\mathrm{PCA}} + h_g^3)\,, \end{aligned} \tag{3.1}$$

after an orthogonal transformation is applied to $(\tilde{\phi}_{k1}, \ldots, \tilde{\phi}_{kd})$, that is, $(\tilde{\phi}_{k1}, \ldots, \tilde{\phi}_{kd})R_0$ for $R_0 \in \mathcal{O}(d)$, if necessary.

(II) The geodesic distance $\hat{d}_g$ estimated by FPTU approximates the true geodesic distance d_g on $\mathcal{M}$ in the sense that, for any geodesic path $(i = i_0, i_1, \ldots, i_{m-1}, i_m = j)$ in G, assuming that $d_g(X_{i_{s-1}}, X_{i_s}) = O(h_g)$ for all $s = 1, \ldots, m$ with some $h_g > 0$, we have

$$\hat{d}_g(\hat{X}_i, \hat{X}_j) = d_g(X_i, X_j) + O_p\{m^2(h_{\mathrm{PCA}} + h_g^3 + h_g h_d^2 \kappa_s)\}\,. \tag{3.2}$$

Theorem 1 includes two parts, and the proofs are given in Supplement B of the Supplementary Material. The first part shows that for two points that are sufficiently close, the orthogonal matrix $\hat{R}_{j,i}$ defined in (2.3) approximates the parallel transport operator $\mathcal{P}_{j,i}$. The second part establishes the consistency of the geodesic distance estimation of FPTU. Let $\{\phi_{\ell 1}, \ldots, \phi_{\ell d}\}$ be an orthonormal basis of $T_\ell\mathcal{M}$. On the right-hand side of (3.1), $\tilde{\phi}_{\ell s}$ can be replaced with $\phi_{\ell s}$, for $s = 1, \ldots, d$; see Supplement B of the Supplementary Material for a proof. The convergence rates are, in general, slower than those under the error-free

high-dimensional data setting (Singer and Wu (2012); Budninskiy et al. (2019)), which depend on whether the points are close to the boundary of $\mathcal{M}$. For points away from the boundary of $\mathcal{M}$, the convergence rates can be faster. In contrast, our convergence rates behave as if all the points are near the boundary of $\mathcal{M}$ in the error-free high dimensional case, although we do choose the rate-optimal bandwidth $h \asymp J^{-1/5}$ to estimate the functional data X. In the case that $\mathcal{M}$ has no boundary, our convergence rates are strictly slower than those in Budninskiy et al. (2019). For any two given points X_i and X_j, the number of vertices of the geodesic path $m \to \infty$ and $mh_g \asymp d_g(X_i, X_j)$, so mh_g does not converge to zero as $n \to \infty$. Therefore, additional rate requirements on h_{PCA} and h_d are needed for the O_p term in (3.2) tending to zero.

Remark 3 (The effects of K and K_{PCA}). Although K and K_{PCA} do not appear explicitly in Theorem 1, their effects on the asymptotic results are reflected through h_g and h_{PCA}, respectively. Recall that $h_{\mathrm{PCA}} = O\{(K_{\mathrm{PCA}}/n)^{1/d}\}$, and that h_g and K have the same relationship. This can be seen by noting that h_g is the bound of the geodesic distance of two adjacent points of a geodesic path in G, whereas G is constructed using K-NN based on the L^2 distance, which is an approximation of the geodesic distance for sufficiently close points. It follows that $h_g = O\{(K/n)^{1/d}\}$ by the same reasoning for h_{PCA} and K_{PCA}.

4. Clustering

As an application of functional manifold learning, we develop a clustering strategy based on the manifold learning outcomes. Let $\mathcal{G} = \{1, \ldots, g\}$ be a collection of group labels with a user-specified number of clusters g. The goal of clustering is to assign each individual to a group in $\mathcal{G}$ such that individuals in the same group are more similar to each other than they are to those in other groups. Recall from Section 2.2 that we first construct a proximity graph G based on K-NN neighborhoods. Provided that K is properly chosen, G may be disconnected, owing to presence of outliers and clusters. Outliers should be removed prior to the implementation of clustering, which can be achieved by discarding components that are too small, for example, those containing only one or two elements. Depending on the practitioner's preference, the removed individuals may be assigned to the closest groups based on the L^2 distance in the end. After outliers are removed, each connected component naturally serves as a first-step cluster. Supposing there are n_c connected components of G, we define the first-step clusters as the connected components. If $n_c = g$, the clustering task is accomplished; if $n_c \neq g$, the second step is needed.

If $n_c > g$, we apply hierarchical clustering by successively merging the two closest connected components of G until the number of clusters reduces to g. Specifically, let C_j be the index set of a connected component of G, for $j = 1, \ldots, n_c$. For all $j \neq j'$, we compute the average-linkage distance (ALD), defined

in Saxena et al. (2017), between C_j and $C_{j'}$,

$$\mathrm{ALD}(C_j, C_{j'}) = \frac{1}{|C_j| \cdot |C_{j'}|} \sum_{i \in C_j} \sum_{i' \in C_{j'}} \|\hat{X}_i - \hat{X}_{i'}\|_{L^2},$$

where $|\{\cdot\}|$ denotes the number of elements in the set $\{\cdot\}$. We merge the two clusters with the smallest ALD, and repeat the above procedure until we have g clusters.

It is more common to encounter the case of $n_c < g$ (n_c is often equal to one, i.e., G is connected). In this case, we perform k-means clustering, with $k = g - n_c + 1$, on the largest component of G using the low-d representations $\hat{Z}_i$ learnt from functional Isomap or FPTU. Specifically, let $\mathcal{S} = \{S_1, \ldots, S_{g-n_c+1}\}$ be a partition of the largest component of G. We search for $\mathcal{S}$ to minimize

$$\sum_{k=1}^{g-n_c+1} \sum_{i \in S_k} \|\hat{Z}_i - \bar{Z}_k\|^2, \tag{4.1}$$

where $\bar{Z}_k = \sum_{i \in S_k} \hat{Z}_i / |S_k|$. The resulting clusters, together with other connected components of G, form our final clustering result. If some outliers have been removed, but still need to be clustered, we can simply assign them individually to the groups that include the individuals closest (based on the L^2 distance) to the removed ones.

When G is connected, that is, $n_c = 1$, our procedure is simply the standard g-means clustering applied on $\hat{Z}_i$. Because the Euclidean distances between the $\hat{Z}_i$ approximate the geodesic distances of our functional data X_i, the rationale of our procedure is the same as that of g-means clustering, but with the Euclidean distance replaced with the geodesic distance. Our method is expected to outperform classical linear methods if the data indeed lie on a curved manifold.

5. Simulations

We compare the numerical performance of the geodesic distance estimation of functional Isomap (FIsomap, see Section 2.2) and the FPTU developed in Section 2.3. We also consider a method that first performs the global functional PCA, and then implements the PTU proposed by Budninskiy et al. (2019) on the principal scores. We choose the number of PCs in the first step to be $\max\{d_{95}, d\}$, where d_{95} is the number of PCs explaining at least 95% of the variance, and d is the intrinsic dimension used in PTU. We refer to the method as $\mathrm{PTU}_{\mathrm{PCA}}$. As in Remark 2, if we rescale v_{i_k}, our method is denoted by $\mathrm{FPTU}_{\mathrm{r}}$; otherwise, it is a non-rescaled version, denoted by $\mathrm{FPTU}_{\mathrm{nr}}$. Similarly, we use $\mathrm{PTU}_{\mathrm{PCA,r}}$ and $\mathrm{PTU}_{\mathrm{PCA,nr}}$ to denote the rescaled and non-rescaled versions of $\mathrm{PTU}_{\mathrm{PCA}}$, respectively. In addition, we compare the clustering strategy based on functional manifold learning introduced in Section 4 with existing methods. We

fix $K_{\mathrm{PCA}} = K$ in our numerical study, because different values of K_{PCA} around K have little effect on the performance of FPTU. To compute the ridged local linear estimator in (2.2), we use a standard Gaussian density $\mathcal{K}$ and a plug-in bandwidth h (Ruppert, Sheather and Wand (1995)), unless otherwise specified.

5.1. Geodesic distance estimation

We consider a unit sphere $\mathbb{S}^2 \subset \mathbb{R}^3$, which is one of the simplest curved manifolds. Let $\theta \in [0, \pi]$ be the latitude and $\phi \in [0, 2\pi]$ be the longitude. A point on $\mathbb{S}^2$ can be parametrized by $(\sin(\theta)\cos(\phi), \sin(\theta)\sin(\phi), \cos(\theta))^T$. Let z_1 and z_2 be any two points on $\mathbb{S}^2$. The geodesic distance between z_1 and z_2 is given by $d_g(z_1, z_2) = 2\arcsin(\|z_1 - z_2\|/2)$. We consider a model of functional data defined by an isometric embedding of $\mathbb{S}^2$ into $L^2([0, 1])$,

$$X(t; \theta, \phi) = \sqrt{2}\{\sin(\theta)\cos(\phi)\sin(2\pi t) + \sin(\theta)\sin(\phi)\cos(2\pi t) + \cos(\theta)\sin(4\pi t)\}, \tag{5.1}$$

for $t \in [0, 1]$. The geodesic distance of two functions X_1 and X_2 is equal to the geodesic distance of the two corresponding points on $\mathbb{S}^2$ induced from $\mathbb{R}^3$.

We consider both error-free and error-prone cases. Let $(0 = t_1, \ldots, t_J = 1)$ be J equi-distant points on $[0, 1]$. We generate the observations $(t_j, Y_{i,j})$ following (5.1); that is, for $i = 1, \ldots, n$ and $j = 1, \ldots, J$, $Y_{i,j} = X_i(t_j; \theta_i, \phi_i) + \epsilon_{i,j}$, where $\theta_i \sim U[0, \pi]$ and $\phi_i \sim U[0, 2\pi]$. In the error-free case, we set $J = 200$, $\epsilon_{i,j} = 0$, and $n = 100, 200$, and 500, for which no smoothing is needed, and we take $\hat{X}_i(t_j) = Y_{i,j}$, for $i = 1, \ldots, n$ and $j = 1, \ldots, J$. In the error-prone case, we set $(J, n) = (30, 100), (60, 200)$, and $(100, 500)$, and $\epsilon_{i,j} \sim N(0, \hat{V}_X/R)$, where $\hat{V}_X$ is the sample variance of X integrated on $[0, 1]$, and $R = 3$ or 10 is the signal-to-noise ratio. For each i, the ridged local linear estimator is applied to $(t_j, Y_{i,j})$ to produce a smooth curve, denoted by $\hat{X}_i$.

We evaluate the performance of the geodesic distance estimation using the mean relative error: $\mathrm{MRE} = \sum_{(i,j)\in\mathbb{N}} |\hat{d}_g(\hat{X}_i, \hat{X}_j) - d_g(X_i, X_j)|/\{d_g(X_i, X_j)|\mathbb{N}|\}$, where $\hat{d}_g$ denotes any estimator of geodesic distance produced by $\mathrm{FPTU_r}$, $\mathrm{FPTU_{nr}}$, $\mathrm{PTU_{PCA,r}}$, $\mathrm{PTU_{PCA,nr}}$, or FIsomap, and $\mathbb{N}$ is the set of all pairs of connected points for a given sample. For FPTU and $\mathrm{PTU_{PCA}}$, the intrinsic dimension $d = 2$ is assumed to be known, or is estimated using the method introduced in Section 2.4. We explore the combinations of $(n, K = K_{\mathrm{PCA}})$ as $(100, 10)$, $(200, 13)$, and $(500, 18)$. Slightly different choices of K and K_{PCA} give similar results, and so are not shown here. For each setting, we apply all methods to 100 random samples. Table 1 summarizes the means and standard deviations of the MREs $\times 10^2$ under different configurations.

As shown in Table 1, both $\mathrm{FPTU_r}$ and $\mathrm{FPTU_{nr}}$ significantly outperform FIsomap in almost all settings, except when d is estimated and the noise level is moderate ($R = 10$). $\mathrm{PTU_{PCA}}$ performs worse than FPTU in almost all error-

Table 1. Mean (standard deviation) of MREs $\times 10^2$ for geodesic distance estimation with 100 random samples.

Methods			FPTU_r	FPTU_nr	$\text{PTU}_{\text{PCA,r}}$	$\text{PTU}_{\text{PCA,nr}}$	FIsomap
$n = 100$	Error-free	d known	2.94 (2.59)	3.95 (2.12)	2.96 (2.50)	3.96 (2.04)	14.74 (6.85)
		d estimated	3.58 (3.70)	4.61 (3.69)	3.59 (3.64)	4.63 (3.66)	
	$R = 10$	d known	14.30 (2.55)	18.20 (1.58)	15.65 (1.80)	18.64 (1.52)	19.91 (6.63)
		d estimated	19.45 (4.72)	22.45 (3.72)	22.04 (5.33)	23.26 (3.89)	
	$R = 3$	d known	22.49 (3.36)	22.74 (1.97)	20.40 (2.65)	23.19 (2.27)	32.79 (7.25)
		d estimated	24.94 (3.43)	26.23 (2.85)	27.11 (3.92)	27.78 (2.95)	
$n = 200$	Error-free	d known	0.92 (0.36)	1.99 (0.22)	0.93 (0.36)	2.00 (0.23)	7.67 (1.99)
		d estimated	1.03 (1.14)	2.09 (1.00)	1.04 (1.15)	2.09 (1.01)	
	$R = 10$	d known	9.08 (1.21)	11.98 (0.75)	9.96 (0.79)	12.39 (0.78)	13.71 (2.56)
		d estimated	12.54 (4.64)	15.23 (4.23)	14.39 (5.66)	15.87 (4.48)	
	$R = 3$	d known	16.68 (2.00)	15.88 (1.11)	14.05 (0.96)	16.61 (0.91)	25.49 (3.51)
		d estimated	18.84 (2.37)	19.83 (2.44)	21.67 (4.24)	22.27 (3.15)	
$n = 500$	Error-free	d known	0.33 (0.04)	1.18 (0.03)	0.33 (0.04)	1.19 (0.03)	3.35 (0.41)
		d estimated	0.33 (0.04)	1.18 (0.03)	0.33 (0.04)	1.19 (0.03)	
	$R = 10$	d known	7.50 (0.55)	8.71 (0.38)	7.42 (0.32)	9.12 (0.34)	11.29 (0.98)
		d estimated	11.85 (2.86)	13.81 (3.26)	15.42 (5.06)	15.90 (4.29)	
	$R = 3$	d known	15.15 (0.87)	12.50 (0.59)	11.17 (0.50)	13.29 (0.52)	22.68 (1.46)
		d estimated	15.74 (1.07)	16.71 (1.04)	21.02 (1.12)	21.04 (0.95)	

prone settings, except for a few cases where d is known and the noise level is high ($R = 3$). These results support the effectiveness of FPTU. Note that PTU_PCA and FPTU perform similarly in the error-free settings. This is as expected, because the global functional PCA used in PTU_PCA captures all information of the functional data in the error-free settings. Furthermore, FPTU_r outperforms FPTU_nr, except in the case of large noise ($R = 3$) and known d. As discussed in Remark 2, using a non-rescaled v_{i_k} (i.e., FPTU_nr) mitigates the distortion caused by large noise. However, for small or no noise, using a rescaled v_{i_k} (i.e., FPTU_r) yields more accurate results.

5.2. Clustering

We compare the clustering strategy using the functional manifold learning outcomes introduced in Section 4 with several existing methods: k-means, the standard k-means clustering applied on full curves with the L_2 distance; k-means_PCA, the standard k-means clustering applied on the PC scores of the curves obtained using a functional PCA with the Euclidean distance, where the number of PCs is chosen to explain at least 95% of the variance; and the projection method developed by Delaigle, Hall and Pham (2019), abbreviated as DHP. For DHP, we use the code from `https://researchers.ms.unimelb.edu.au/~aurored/` with $\rho = 0.2$, the Haar basis functions, and the number of basis functions equal to four. A larger number of basis functions does not significantly improve the performance for our examples, but does increase the computation cost.

We cluster individuals into two groups (the ground truth). Let $(0 = t_1, \ldots, t_J$

$= 1$) be J equi-distant points on $[0, 1]$. The observed data $(t_j, Y_{i,j})$, for $i = 1, \ldots, n$ and $j = 1, \ldots, J$, are generated from the following models, which are variants of common examples in the manifold learning literature:

(i) Skewed Swiss roll:

$$Y_{i,j} = Z_{1i}\cos(Z_{1i})\sin(2\pi t_j) + \{Z_{1i}\sin(Z_{1i}) + Z_{2i}\}\cos(2\pi t_j) + Z_{2i}\sin(4\pi t_j) + \epsilon_{i,j}\,,$$

where $Z_{1i} \sim U[0, 7]$ if i is in group one, $Z_{1i} \sim U[7, 10]$ if i is in group two, and $Z_{2i} \sim U[0, 4]$.

(ii) S shape:

$$Y_{i,j} = \sin(Z_{1i})\sin(2\pi t_j) + \text{sign}(Z_{1i})\{\cos(Z_{1i}) - 1\}\cos(2\pi t_j) + Z_{2i}\sin(4\pi t_j) + \epsilon_{i,j}\,,$$

where $Z_{1i} \sim U[-3\pi/2, 0]$ if i is in group one, $Z_{1i} \sim U[0, 3\pi/2]$ if i is in group two, and $Z_{2i} \sim U[1, 4]$.

(iii) Time warping:

$$Y_{i,j} = |Z_{2i}|\mu\{\gamma_i(t_j)\} + \epsilon_{i,j}\,, \text{ with } \mu(t) = \phi_{0.2,0.08}(t) + \phi_{0.5,0.1}(t) + \phi_{0.8,0.13}(t)\,,$$

where $\gamma_i(t) = \{\exp(Z_{1i}t) - 1\}/\{\exp(Z_{1i}) - 1\}$ if $Z_{1i} \neq 0$, and $\gamma_i(t) = t$ if $Z_{1i} = 0$. Here, $Z_{1i} \sim U[-1, 0]$ if i is in group one, $Z_{1i} \sim U[0, 1]$ if i is in group two, $Z_{2i} \sim N(1, 0.1)$, and $\phi_{\mu,\sigma}$ is the probability density function of $N(\mu, \sigma^2)$.

For all of the aforementioned models, $\epsilon_{i,j} \sim N(0, \hat{V}_X/R)$ with $\hat{V}_X$, the sample variance of X integrated on $[0, 1]$, and $R = 10$ to induce a moderate noise level. Models (i) and (ii) are embeddings from well-known manifold examples used in high-dimensional data settings (e.g., Tenenbaum, De Silva and Langford. (2000); Ma and Fu (2012)) into $L^2([0, 1])$. The skewed Swiss roll in model (i) is slightly different from the standard one to make it more challenging. Model (iii) is a time warping model showing the significant phase variation ubiquitous in functional data, which is a classical type of manifold for functional data (Srivastava et al. (2011); Chen and Müller (2012)). The intrinsic dimension d of these models is two. For the functional manifold learning methods, we assume d is either known or we estimate it using the method in Section 2.4.

We set $(J, n) = (60, 300)$ and $(100, 500)$, and use $K = K_{\text{PCA}} = 13, 15$, and 18 for each case. Half of the individuals are assigned to group one, and the other half forms group two. Before we apply the clustering methods to the generated data, we use the ridged local linear estimator to obtain the smooth curves $\hat{X}_i$. Following the strategy in Section 4, if the proximity graph G is not connected, the connected components of G with size less than three are discarded as outliers. The remaining individuals are clustered and the assessment metric is the adjusted

Rand index (ARI, Hubert and Arabie (1985)). The ARI measures the similarity of two partitions of the data. A larger ARI compared with the ground truth means a better clustering result, and the maximum value is one. We replicate 100 samples under each model; the results are summarized in Table 2.

Table 2 shows that the clustering strategies using FPTU and FIsomap outperform the other methods in all settings, especially under model (i). PTU_{PCA} performs poorly under model (i), but similarly to FPTU under models (ii) and (iii). Among FPTU_{r}, FPTU_{nr}, and FIsomap, they are quite competitive overall: FPTU_{r} and FPTU_{nr} perform slightly better than FIsomap under model (i), but their performance under model (ii) is affected significantly by the estimation of d. Unlike FIsomap, FPTU_{r} and FPTU_{nr} use d in the step of geodesic distance estimation, which could be a drawback if d is estimated poorly.

6. Real-Data Examples

6.1. Berkeley growth data

The Berkeley growth data set (Tuddenham and Snyder (1954)) is a classical functional data example that has been studied extensively in the literature. A notable feature of growth data is that different individuals often possess significant phase variation, which hinders a standard linear analysis (e.g., FPCA) if phase variation is ignored (Ramsay and Silverman (2005)). Chen and Müller (2012) formalized phase variation using manifold terminology, and investigated part of this data set in the setting of functional manifold learning.

The data set includes height measurements for 39 boys and 54 girls from age 1 to 18. Gender can be used as a cluster benchmark for assessing clustering methods, that is, the clustering ground truth is known. To show distinct phase variation, we use the ridged local linear estimator to obtain the first derivatives of the growth data, that is, the growth velocity curves, as shown in the left panel of Figure 1. We then apply each clustering method to the growth velocity curves, and assess the results using the ARI for gender clusters. For the manifold learning methods, we explore the choices of $d = 2, 4$, and 6 and $K = K_{\text{PCA}} = 8, 10$, and 12. With such values of K, the proximity graph G includes a few singleton components, which are first removed, and then later assigned to the closest groups based on the L^2 distance.

Table 3 shows the clustering results assessed using the ARI for all clustering methods. We can see that DHP performs best in terms of distinguishing gender clusters, followed by our manifold learning methods. Different choices of d and K do not affect the results significantly. The linear methods, namely, $k\text{-means}_{\text{PCA}}$ and k-means, perform worst in this example. To visualize the differences between the functional manifold learning methods and the standard FPCA, we show the manifold learning outcomes $\hat{Z}_i$ obtained by performing MDS on all the geodesic distance matrices and the principle scores of the first two principle components

of the FPCA in Figure 2. We see that a few mis-clustered girls ($*$) are clearly within the boy ($\circ$) cluster produced by the manifold learning methods, and that all of the mis-clustered individuals are near the boundary of the gender clusters of the point cloud produced by the FPCA. Of the manifold learning methods, FIsomap yields the largest scale of the points, and $\text{PTU}_{\text{PCA,nr}}$ yields the smallest scale. This is as expected, because the length information contained in the higher (both global and local) principle components, for example, larger distances caused by noise, is ignored in the procedure of $\text{PTU}_{\text{PCA,nr}}$.

6.2. Yeast gene expression data

The second example focus on gene expression data of a yeast cell (Spellman et al. (1998)), where we conduct an α factor-based synchronization experiment. A total of 6178 genes were measured every seven minutes, 18 times, among which 612 genes were identified as being periodic and recorded without missing values. These periodic genes were classified into five groups: G_1, G_2, M, M/G_1, and S. However, Zhao, Marron and Wells (2004) suggested that a large number of genes are not periodic, and Leng and Müller (2006a) and Leng and Müller (2006b) studied only a subset of 89 genes. To illustrate the clustering methods, we focus on the groups G_1, G_2 and M; the components of the proximity graph G (using $K = 10$) with sizes less than six are discarded as outliers. This results in a sample of size 427, among which 195 individuals in group G_1 form one cluster, and 232 individuals in groups G_2 and M belong to another cluster. They serve as the ground truth of the clustering for this example.

Because the raw data include significant noise and the number of measurements is moderate, we use a local linear estimator to smooth the data, with a manually chosen large bandwidth of five (note that the time domain is $[0, 119]$). The resulting curves are depicted in the right panel of Figure 1. We apply the clustering methods to the smoothed data, and evaluate their performance using the ARI. We choose $d = 4, 6$, and 8 and $K = K_{\text{PCA}} = 12, 14$, and 16 for the manifold learning methods, and summarize the clustering results in Table 4.

Table 4 shows that FPTU_{r} and FPTU_{nr} with $d = 4$ and $K = 12$ perform best, and that $k\text{-means}_{\text{PCA}}$ and k-means are also quite competitive. PTU_{PCA} produces a few disconnected components of the proximity graph when $d = 4$ and $K = 12$ and 14, so its performance is poor. In the cases of $d = 6$ and 8, PTU_{PCA} selects d PCs of the global functional PCA in the first step, and thus PTU simply uses the Euclidean distances of the principal scores in the second step. This is why PTU_{PCA} produces the same value for ARI for all cases of $d = 6$ and 8. DHP does not work well in terms of identifying the predefined clusters for this data set. FIsomap, which perform similarly to FPTU_{r} and FPTU_{nr} in Sections 5.2 and 6.1, performs much worse than these two for this data set. This is probably because the large noise makes the geodesic distance estimates poor under FIsomap. Our proposed FPTU is more robust to large noise.

Table 2. Simulation results of clustering with the mean of the ARIs $\times 10^2$ computed from 100 samples.

$(J,n)=(60,300)$		FPTU_r			FPTU_nr			$\text{PTU}_{\text{PCA,r}}$			$\text{PTU}_{\text{PCA,nr}}$			FIsomap			DHP	k-means$_{\text{PCA}}$	k-means
K		13	15	18	13	15	18	13	15	18	13	15	18	13	15	18	–	–	–
Model (i)	d known	90.17	89.94	91.38	90.27	90.47	91.86	17.75	17.63	17.59	17.63	17.66	17.66	87.93	88.07	89.02	42.99	17.98	17.18
	d estimated	90.09	89.36	90.94	90.29	89.87	91.53	17.73	17.57	17.78	17.17	17.59	17.77	87.93	88.07	89.03			
Model (ii)	d known	93.07	93.05	92.14	93.06	92.96	92.20	93.69	93.36	92.64	93.69	93.44	92.84	92.83	93.17	92.93	8.74	86.65	86.51
	d estimated	88.62	88.50	89.03	88.78	88.99	89.26	89.28	89.11	89.23	89.36	89.21	89.26	92.84	93.16	92.83			
Model (iii)	d known	87.02	86.98	86.76	85.64	85.58	85.35	84.35	84.99	85.22	84.20	85.14	84.91	86.37	86.48	86.52	48.95	82.54	82.48
	d estimated	87.03	86.87	86.68	85.58	85.46	85.47	82.29	82.81	82.95	82.36	83.06	82.93	86.36	86.42	86.44			
$(J,n)=(100,500)$		FPTU_r			FPTU_nr			$\text{PTU}_{\text{PCA,r}}$			$\text{PTU}_{\text{PCA,nr}}$			FIsomap			DHP	k-means$_{\text{PCA}}$	k-means
K		13	15	18	13	15	18	13	15	18	13	15	18	13	15	18	–	–	–
Model (i)	d known	88.89	88.91	88.58	88.79	88.92	89.34	19.19	19.33	19.27	18.92	19.58	19.48	86.88	87.22	87.31	44.65	19.49	19.16
	d estimated	88.86	88.87	88.58	88.79	88.93	89.33	18.85	19.81	19.39	18.97	19.37	19.54	86.85	87.22	87.31			
Model (ii)	d known	95.00	95.19	95.13	95.25	95.20	95.41	94.99	95.09	95.32	95.19	95.39	95.55	94.69	95.02	95.07	8.48	86.42	86.19
	d estimated	91.25	89.99	88.42	91.68	90.34	89.14	88.62	88.71	88.70	88.69	88.60	88.85	94.69	95.01	95.11			
Model (iii)	d known	88.45	88.19	89.05	87.09	86.96	87.69	83.42	86.83	87.46	83.37	86.72	87.51	88.12	87.92	88.94	61.40	85.34	85.33
	d estimated	87.98	88.11	88.82	86.96	86.89	87.73	81.58	84.97	85.76	81.54	84.93	85.70	87.99	87.90	88.97			

Table 3. The ARIs $\times 10^2$ for all methods applied to Berkeley growth data.

		FPTU_r			FPTU_nr			$\text{PTU}_{\text{PCA,r}}$			$\text{PTU}_{\text{PCA,nr}}$			FIsomap			DHP	k-means$_{\text{PCA}}$	k-means
K		8	10	12	8	10	12	8	10	12	8	10	12	8	10	12	–	–	–
d	2	71.86	71.86	61.19	68.21	71.86	54.55	68.21	61.19	64.66	68.21	64.66	64.66	68.21	68.21	64.66	87.38	57.84	57.84
	4	64.66	64.66	64.66	64.66	68.21	64.66	64.65	64.66	61.20	64.65	64.66	61.20	71.86	68.21	64.66			
	6	64.66	64.66	61.20	64.66	64.66	61.20	68.21	61.20	64.66	68.21	61.20	64.66	71.86	68.21	64.66			

Table 4. The ARIs $\times 10^2$ for all methods applied to the yeast gene expression data.

		FPTU_r			FPTU_nr			$\text{PTU}_{\text{PCA,r}}$			$\text{PTU}_{\text{PCA,nr}}$			FIsomap			DHP	k-means$_{\text{PCA}}$	k-means
K		12	14	16	12	14	16	12	14	16	12	14	16	12	14	16	–	–	–
d	4	73.01	70.62	68.27	76.25	73.81	65.96	-0.07	-0.07	71.41	-0.07	-0.07	73.01	39.53	40.72	33.83	33.80	67.49	66.73
	6	67.49	63.69	62.20	65.20	69.05	65.96	66.73	66.73	66.73	66.73	66.73	66.73	38.94	40.12	34.37			
	8	67.49	67.49	64.44	65.96	67.49	64.44	66.73	66.73	66.73	66.73	66.73	66.73	38.94	40.12	36.05			

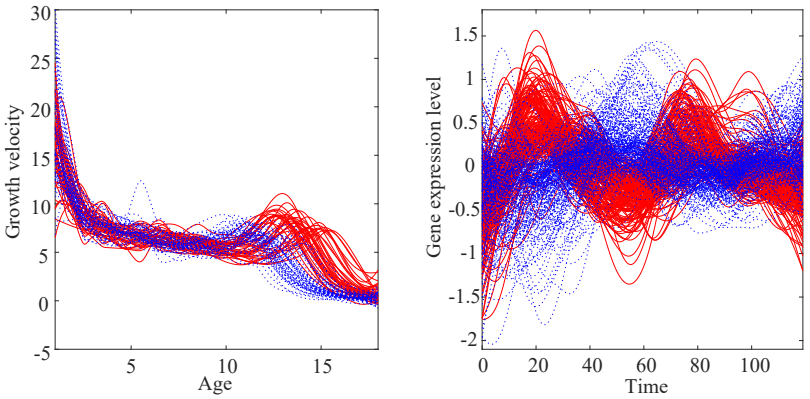

Figure 1. Left: Berkeley growth velocity data, where a solid line (−) denotes boy and a dotted line (· · ·) denotes girl. Right: Yeast gene expression data, where a solid line (−) denotes group G_1 and a dotted line (· · ·) denotes groups G_2 and M.

7. Discussion

In manifold learning for functional data, we adapt the well-known Isomap into our setting, and develop a functional version of parallel transport unfolding to produce a more robust geodesic distance estimation. We derive the asymptotic convergence rates of FPTU, showing that they are slower than their high-dimensional counterparts, in general, owing to the noise. Using functional manifold learning outcomes, we propose a graph-based clustering strategy, and show using several synthetic examples that our proposed strategy outperforms others if the data indeed lie on a curved manifold. When applied to real-data examples, the functional manifold learning techniques reveal features different from those of the standard method, as shown by Figure 2.

Note that our setting is quite different from that of manifold-valued functional data, where the manifold is usually known *a priori*. It would be interesting to extend the idea of manifold learning to the case of functional data valued in an unknown manifold.

Classical manifold learning techniques have had tremendous success in pattern recognition, and mitigate the curse of dimensionality for high-dimensional data. The ideas behind such techniques have been used to identify manifold representations (Chen and Müller (2012)) and nonparametric regression (Lin and Yao (2020)) in functional data analysis. Here, we use functional manifold learning to conduct clustering, based on the hypothesis that the clusters lie on different parts of an unknown manifold. It is of interest to explore other applications of functional manifold learning, for example, whether the assumption that the data lie on an unknown manifold is useful in dealing with sparsity, missingness, and noise in functional data.

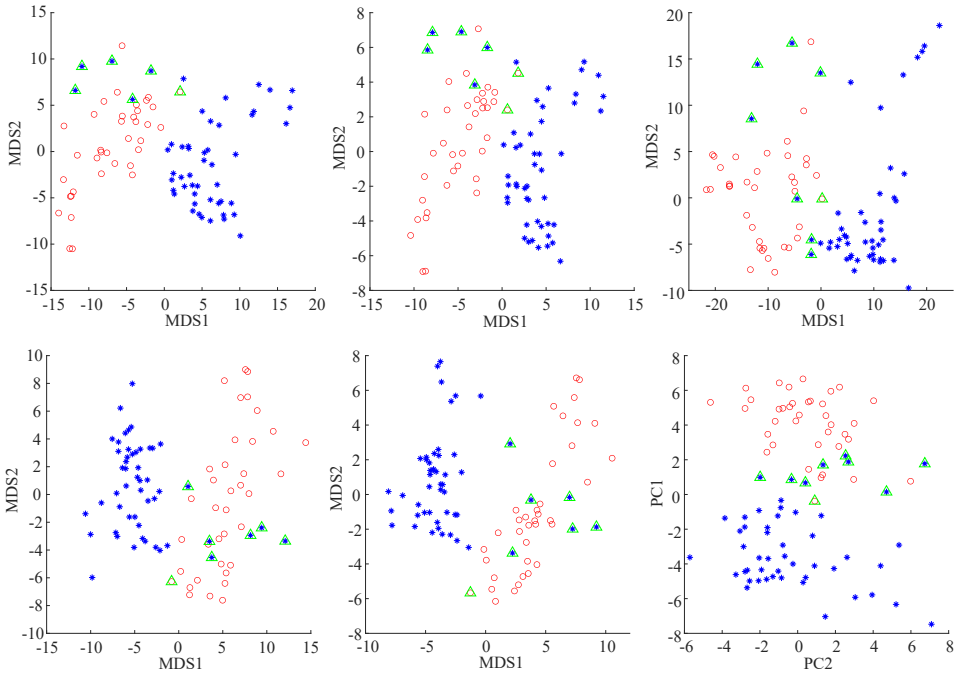

Figure 2. The manifold learning outcomes $\hat{Z}_i$ (the two coordinates from MDS) with $d = 2$ and $K = 8$ using FPTU_{r} (top-left), FPTU_{nr} (top-middle), FIsomap (top-right), $\text{PTU}_{\text{PCA,r}}$ (bottom-left), and $\text{PTU}_{\text{PCA,nr}}$ (bottom-middle). The bottom-right panel shows the scores of the first two principle components (PC1 and PC2) using FPCA. Boys are denoted by $\circ$ and girls by $*$. Mis-clustered individuals are highlighted by $\triangle$.

Supplementary Material

The online Supplementary Material contains technical proofs. The code to reproduce the simulations is available at `https://github.com/ruoxut/FunctionalManifoldLearning`.

Acknowledgments

We thank the two referees for their careful reviews of our manuscript and insightful comments. This research was partially supported by funding from the Research Grants Council of Hong Kong (17308321).

References

Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.* **15**, 1373–1396.

Budninskiy, M., Yin, G., Feng, L., Tong, Y. and Desbrun, M. (2019). Parallel transport unfolding: A connection-based manifold learning approach. *SIAM Journal on Applied Algebra and Geometry* **3**, 266–291.

Chen, D. and Müller, H.-G. (2012). Nonlinear manifold representations for functional data. *Ann. Statist.* **40**, 1–29.

Cox, M. A. and Cox, T. F. (2008). Multidimensional scaling. In *Handbook of Data Visualization*, 315–347. Springer.

Dai, X., Lin, Z. and Müller, H.-G. (2020). Modeling sparse longitudinal data on Riemannian manifolds. *Biometrics* **77**, 1328–1341.

Dai, X. and Müller, H.-G. (2018). Principal component analysis for functional data on Riemannian manifolds and spheres. *Ann. Statist.* **46**, 3334–3361.

Delaigle, A., Hall, P. and Pham, T. (2019). Clustering functional data into groups by using projections. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **81**, 271–304.

Facco, E., d'Errico, M., Rodriguez, A. and Laio, A. (2017). Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Sci. Rep.* **7**, 1–8.

Fan, J. and Gijbels, I. (1996). *Local Polynomial Modelling and Its Applications: Monographs on Statistics and Applied Probability 66*. CRC Press.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *J. Classification* **2**, 193–218.

Kneip, A. and Gasser, T. (1992). Statistical tools to analyze data representing a sample of curves. *Ann. Statist.* 1266–1305.

Leng, X. and Müller, H.-G. (2006a). Classification using functional data analysis for temporal gene expression data. *Bioinformatics* **22**, 68–76.

Leng, X. and Müller, H.-G. (2006b). Time ordering of gene coexpression. *Biostatistics* **7**, 569–584.

Lila, E., Aston, J. A. and Sangalli, L. M. (2016). Smooth principal component analysis over two-dimensional manifolds with an application to neuroimaging. *Ann. Appl. Stat.* **10**, 1854–1879.

Lin, Z., Shao, L. and Yao, F. (2020). Intrinsic riemannian functional data analysis for sparse longitudinal observations. *arXiv:2009.07427*.

Lin, Z. and Yao, F. (2019). Intrinsic Riemannian functional data analysis. *Ann. Statist.* **47**, 3533–3577.

Lin, Z. and Yao, F. (2020). Functional regression on the manifold with contamination. *Biometrika* **108**, 167–181.

Ma, Y. and Fu, Y. (2012). *Manifold Learning Theory and Applications*. CRC Press, Boca Raton.

Pless, R. and Souvenir, R. (2009). A survey of manifold learning for images. *IPSJ Transactions on Computer Vision and Applications* **1**, 83–94.

Ramsay, J. and Silverman, B. W. (2005). *Functional Data Analysis*. Springer.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science* **290**, 2323–2326.

Ruppert, D., Sheather, S. J. and Wand, M. P. (1995). An effective bandwidth selector for local least squares regression. *J. Amer. Statist. Assoc.* **90**, 1257–1270.

Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O. P., Tiwari, A. et al. (2017). A review of clustering techniques and developments. *Neurocomputing* **267**, 664–681.

Singer, A. and Wu, H.-T. (2012). Vector diffusion maps and the connection Laplacian. *Comm. Pure Appl. Math.* **65**, 1067–1144.

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B. et al. (1998). Comprehensive identification of cell cycle–regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell* **9**, 3273–3297.

Srivastava, A., Wu, W., Kurtek, S., Klassen, E. and Marron, J. S. (2011). Registration of functional data using Fisher-Rao metric. *arXiv:1103.3817*.

Tenenbaum, J. B., De Silva, V. and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science* **290**, 2319–2323.

Tuddenham, R. D. and Snyder, M. M. (1954). Physical growth of California boys and girls from birth to age 18. In *University of California Publications in Child Development*, 183–364. University of California Press.

Yao, F., Müller, H.-G. and Wang, J.-L. (2005). Functional data analysis for sparse longitudinal data. *J. Amer. Statist. Assoc.* **100**, 577–590.

Zhang, Z. and Saparbayeva, B. (2021). Amplitude mean of functional data on $\mathbb{S}^2$. *arXiv: 2107.13721*.

Zhang, Z. and Zha, H. (2004). Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM Journal on Scientific Computing* **26**, 313–338.

Zhao, X., Marron, J. and Wells, M. T. (2004). The functional data analysis view of longitudinal data. *Statist. Sinica* **14**, 789–808.

Ruoxu Tan

School of Mathematical Sciences, Key Laboratory of Intelligent Computing and Applications (Ministry of Education), Tongji University, Shanghai, China.

E-mail: ruoxut@tongji.edu.cn

Yiming Zang

Université de Lorraine, CNRS, IECL, F-54000 Nancy, France.

E-mail: yiming.zang@univ-lorraine.fr

Guosheng Yin

Department of Mathematics, Imperial College London, London SW7 2AZ, UK.

E-mail: guosheng.yin@imperial.ac.uk

(Received November 2021; accepted October 2022)