Robust Functional Manifold Clustering

Yi Guo, Stephen Tierney, and Junbin Gao

Abstract— **In machine learning, it is common to interpret each data sample as a multivariate vector disregarding the correlations among covariates. However, the data may actually be functional, i.e., each data point is a function of some variable, such as time, and the function is discretely sampled. The naive treatment of functional data as traditional multivariate data can lead to poor performance due to the correlations. In this article, we focus on subspace clustering for functional data or curves and propose a new method robust to shift and rotation. The idea is to define a function or curve and all its versions generated by shift and rotation as an equivalent class and then to find the subspace structure among all equivalent classes as the surrogate for all curves. Experimental evaluation on synthetic and real data reveals that this method massively outperforms prior clustering methods in both speed and accuracy when clustering functional data.**

Index Terms— **Clustering, curves, functional data, manifold.**

I. Introduction

IN MACHINE learning, it is common to interpret each data point as a vector in the Euclidean space [1]. Such a discretization is chosen because it allows for easy manipulations and fast computation, even with large data sets. However, these methods choose to ignore that the data may not naturally fit into this assumption. In fact, much of the data collected for practical machine learning are actually functions or curves. In contrast to feature vectors, functional data encode gradient information, which is vital to analysis. For example, financial data, such as stock or commodity prices, are functions of monetary values over time [2]. Recently, functional data have become increasingly important in many scientific and engineering research areas, such as electrocardiogram (ECG) or electroencephalography (EEG) in healthcare [3], subject outlines in both macrobiology and microbiology [4], weather or climate data [5], astronomy [6], and motion trajectories from computer vision [5], [7].

Analyzing functional data has been an emerging topic in statistical research [8]–[10] and has attracted great attention from machine learning community in recent years [11], [12]. One of the important challenges in analyzing functional data for machine learning is to efficiently cluster and to learn better representations for functional data. Theoretically, functional data are of infinite dimension. Due to discretization, there are only limited samples taken from a functional observation. A strong correlation between neighboring samples and the information loss during discretization are major challenges for analysis. The desired model for functional data is expected to properly and parsimoniously characterize the nature and variability hidden in the data. The classic functional principal component analysis (fPCA) [13], [14] is one of such examples to discover dominant modes of variation in the data. However, fPCA may fail to capture patterns if the functional data are not well aligned in its domain. For time series, a special type of functional data, i.e., dynamic time warping (DTW), has long been proposed to compare time series based on shape and distortions (e.g., shifting and stretching) along the temporal axis [15], [16].

Another important type of functional data is shape [9], [17]. The shape is an important characterizing feature for objects, and in computer vision, the shape has been widely used for the purpose of object detection, tracking, classification, and recognition. In fact, a natural and popular representation for shape analysis is to parameterize the boundaries of planar objects as 2-D curves. In object recognition, images of the same object should be similar regardless of resolution, lighting, or orientation. Hence, an efficient shape representation or shape analysis scheme must be invariant to scale, translation, and rotation.

Our intention in this study is to consider functional data clustering by accounting for the possible invariance in scaling/stretching, translation, and rotation of functional data. The focus of this article is upon functional data, including continuous functions parameterized by a single variable, such as time and shapes in the Euclidean spaces. The main characteristic that we are interested in is that the functions are actually clustered in underlying subspaces. In other words, the original functions are sampled from several subspaces, which are embedded in a space with infinitely many dimensions. However, the observed functions are affected by geometric distortions and noise for some reason. The distortions creep in the multivariate representations of the functions and effectively break down the subspace identification methods built on classic multivariate data, even for those with robustness designed in the models to handle fairly large noises. Our experiments shown in Section V reveal this quite clearly. The reason is straightforward as the geometric distortions are themselves structural and do not fit into any probability distribution.

Manuscript received July 22, 2019; revised December 17, 2019; accepted March 4, 2020. Date of publication April 2, 2020; date of current version February 4, 2021. The work of Junbin Gao was supported by the Australian Research Council (ARC) under Grant DP140102270. *(Corresponding author: Yi Guo.)*

Yi Guo is with the Centre for Research in Mathematics and Data Science, School of Computer, Data and Mathematical Sciences, Western Sydney University, Parramatta, NSW 2150, Australia (e-mail: y.guo@westernsydney.edu.au).

Stephen Tierney and Junbin Gao are with the Discipline of Business Analytics, The University of Sydney Business School, The University of Sydney, Sydney, NSW 2006, Australia (e-mail: stephen.tierney@sydney.edu.au; junbin.gao@sydney.edu.au).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.2979444

The solution to this problem relies on countering distortions, and the way we approach it is to manipulate the observed functions such that the transformed versions are invariant to distortions, or in other words, we group the distorted versions of one function to be an equivalent class and treat the whole class as a datum in clustering. The equivalent class as a new representation accommodates all possible distortions and, hence, an infinite class. However, the cost is that the new representation of functions has some geometric structure, which has to be dealt with to achieve the final goal. This geometric structure, in particular, is a quotient manifold where each point is an equivalent class of functions that are transformed versions of each other. The original subspace then becomes subspaces in this quotient manifold. However, it has no obvious coordinate representation, and hence, no computation can be carried out directly. We get around this problem by projecting points into tangent space, a vector space regarded as a local approximation of the manifold, where we model the relations among the images of the points for clustering purposes. A very useful shape representation we consider is the square-root velocity function (SRVF) representation [9], [18]. In general, the resulting SRVF of a continuous shape is square-integrable, belonging to the well-defined Hilbert space where appropriate measurement can be applied. Refer to Section III for more details. By acknowledging the true nature of the data, we develop a more robust clustering method that exploits features that would otherwise be ignored or lead to erroneous results with simple linear models.

The rest of this article is organized as follows. In Section II, we discuss related work in this field. In Section III, we review the preliminaries about the SRVF and more importantly the manifold of open curves and introduce our robust functional manifold clustering (rFMC) model. Section IV is dedicated to explaining an efficient algorithm for solving the optimization in the realization of our model based on the linearized alternative direction method with an adaptive penalty (LADMAP), and the algorithm convergence and complexity are also analyzed. In Section V, the proposed model is assessed on both synthetic and real-world databases against several state-of-the-art methods. Finally, conclusions are discussed in Section VI.

II. Related Work and Motivation

The simplest approaches for clustering functional data have relied heavily on DTW. DTW is an alignment technique that aims to warp the time axis of the data until the difference between the two sequences is minimized [19]. DTW also provides a distance measurement between the two sequences, once aligned. Historically, DTW has been mainly used for large scale data mining, where queries are performed to quickly find the nearest neighbors to sequences of data [6]. However, the aligned distance produced by DTW has been used for the clustering of functional data. In its simplest form, DTW is used to produce a pairwise distance matrix for the entire data set [20], [21]. Then, the distance matrix is used by a hierarchical or spectral clustering method to produce the final clusters. Although these DTW-based approaches are computationally cheap, their clustering accuracy leaves much to be desired. This is due to two flaws in DTW. First, the distance measurement for DTW is based on the Euclidean distance between each point in the sequence. This totally ignores any gradient-based information in the sequence. Second, warping accuracy is dependent on the correct choice of window size. Poor choices of warping window size can have a dramatic impact on warping and alignment accuracy [22]. Other methods, such as DTW-HMM [23], have used DTW-based clusters as an initialization point for more advanced methods, such as hidden Markov models.

A more sophisticated approach for functional data clustering is to use probabilistic methods [19]. Early methods used simple Gaussian probability models [5], [24]–[26]. However, these models only hold for the Euclidean vector data where the notions of cluster centers and cluster variance can be easily quantified [27]. Zhang *et al.* [28] proposed the Bayesian clustering of curves, which is similar in nature to DTW pairwise distance clustering with advances that address most of the drawbacks. The distance matrix is based on the analysis of the data points in the curve manifold. Then, the clustering is performed on the distance matrix by using a probabilistic method that simultaneously finds the cluster assignment and the number of clusters automatically.

We now draw the readers' attention to subspace clustering methods, as they apply to many situations that traditional clustering methods cannot handle well [29], [30]. Different from finding spatially concentrated clusters measured by the usual Euclidean metric, these methods aim to segment the data into clusters with each cluster corresponding to a unique subspace. More formally, given a data matrix of observed columnwise data samples $\mathbf{A} = [\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_N}] \in \mathbb{R}^{p\times N}$ and $\mathbf{a}_i \in \mathbb{R}^p \ \forall i = 1, \ldots, N$, the objective of subspace clustering is to assign each data sample to its underlying subspace. The basic assumption is the following.

Subspace Assumption: The data are drawn from a union of c subspaces $\{S_i\}_{i=1}^c$ with bases $\{B_i\}_{i=1}^c$ and intrinsic dimensions $\{d_i\}_{i=1}^c$.

Based on the above-mentioned assumption, simple linear algebra shows that for any given point in a subspace, the most parsimonious or consistent (measured by matrix rank) reconstruction of it is formed by the points from the same subspace as that of the given one. Many methods are constructed on this observation, such as sparse subspace clustering (SSC) [31] and low-rank representation (LRR) [32], while different methods seek different structures in the reconstruction matrices in their formulations. For any given datum $\mathbf{a}_i$, the reconstruction is expressed as

$$\mathbf{w}_i = \arg\min_{\mathbf{z}_i} \|\mathbf{a}_i - \mathbf{A}_{-i}\mathbf{w}_i\| \tag{1}$$

where $\mathbf{A}_{-i}$ is the matrix of $\mathbf{A}$ with the ith column removed, and $\mathbf{w}_i$ is a vector with the reconstruction weights for $\mathbf{a}_i$. Let $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_N]$ be the matrix consisting of all reconstruction weights. The restriction to $\mathbf{W}$ as being sparse or low rank has the effect of selecting points from subspaces. Therefore, $\mathbf{W}$ can be used as an affinity matrix to form a graph and a spectral clustering method, such as nCUT [33], to obtain the final subspace labels. Of course, there is always noise in the observed data, which has to be modeled for robustness.

Let us go back to functional data case where $\mathbf{a}_i$ is actually a function of time t without loss of generosity, i.e., $a_i(t) \in \mathbb{R}^\infty$. The subspace assumption can still be valid for the bases being functions as well. This is obvious in spectroscopy where some material spectrum is a mix of several spectra of pure materials [34], which is one set of bases. Segmenting spectra according to their constituents is exactly subspace clustering. Similar applications like this are abundant in many areas, such as computer vision, where the data are essential functions, and therefore, subspace clustering is very useful in a functional setting. As a consequence of the validity of the subspace assumption, the reconstruction in (1) still holds. This justifies the direct application of the subspace clustering to the multivariate representation of functional data when they are clean. However, the observed functionals can be affected by many things except for additive noise. The most serious one is distortions in shape. For example, in thermal infrared data of geological substances, a curve may contain a key identifying feature, such as a dip near a particular frequency. This dip may shift or vary position even for the same substance due to impurities, or in other cases, the feature may be elongated, shrunk, or subject to some nonuniform warping or scaling. Unfortunately, these distortions commonly exist in functional data. For example, the trajectories of writing a letter look very similar but never quite the same with different orientations, starting points, and features; nonetheless, they are the same letter. These distortions effectively break down (1) in an obvious way, e.g., a shifted version of a function cannot even reconstruct its original version using the formulation of (1), and therefore, multivariate subspace clustering is no longer appropriate.

DTW correction can alleviate the problem to some extent, with the limitation of only dealing with shift effectively. Moreover, when two functions are from different subspaces, the optimal alignment is not clear, and the alignment outcome can be misleading as the shape features of functions can be misaligned during the process. Nonetheless, the DTW idea is interesting. What if we take another route? Instead of carrying out the alignment explicitly, we pack all the common distortions into a map $\mathcal{F}$ from a function to its equivalent class, i.e., $\mathcal{F}: a(t) \mapsto [a(t)]$, meaning that if one function is the distorted version of the other, they end up in the same equivalent class by using $\mathcal{F}$. $[x]$ stands for the equivalent class containing all objects that are equivalent to x in some sense. Now, the data in the subspace assumption become equivalent classes instead of individual functions. Although the equivalent class has no explicit vector form, its geometric structure may be used for the computational purpose to recover the subspaces underpinning these equivalent classes. Thus, we need two components at the same time, the map $\mathcal{F}$ accounting for distortions which has a suitable geometric structure. Section III is dedicated to explaining the details of the development of our method.

III. Robust Functional Manifold Clustering

Please note that the differential geometry is involved in this method. The work in [35] is an excellent reference for all the concepts used in this article.

We start with a few observations. The first is that the distinct features of functions are usually in the rate of changes, such as valleys and peaks, which can be better captured by the first derivative. The second is that shift, stretch, and compression of functions can be regarded as reparameterization of the time t, i.e., the distorted version is actually $a(\gamma(\tau))$, where $\gamma(\tau)$ is the reparameterization function replacing t. The third is that the rotation is a simple linear operation whose matrix representation is a rotation matrix. These observations lead us to the SRVF representation [9], [18] with the rotation group detailed in the following.

A. Curve Manifold

We formalize the function as a smooth parameterized n-dimensional curve $a: D \to \mathbb{R}^n$, where $D = [0, 1]$. Here, the smoothness requirement means that the function has continuous first order derivative. We represent it using the SRVF representation given by

$$q(t) = \frac{\dot{a}(t)}{\sqrt{\|\dot{a}(t)\|}}$$

where $\dot{a}(t)$ is the derivative of $a(t)$ and $\|\dot{a}(t)\|$ the L^2 norm of $\dot{a}(t)$. Apparently, we exclude constant curve that has norm 0. The SRVF mapping transforms the original curve $a(t)$ into a gradient-based representation, which facilitates the comparing of the shape information.

In this article, we focus on the set of open curves, i.e., $a(0) \neq a(1)$. For handling general curves, we refer readers to [9]. The SRVF facilitates a measure and geometry bearing invariance to scaling, shifting, and reparameterizing in the curves domain. For example, all the translated curves (in $\mathbb{R}^n$) from a curve $a(t)$ will have the same SRVF. Robinson [36] proved that if the curve $a(t)$ is absolutely continuous, then its SRVF $q(t)$ is square-integrable, i.e., $q(t)$ is in a functional Hilbert space $L^2(D, \mathbb{R}^n)$. Conversely, for each $q(t) \in L^2(D, \mathbb{R}^n)$, there exists a curve $a(t)$ whose SRVF corresponds to $q(t)$. Thus, the set $L^2(D, \mathbb{R}^n)$ is a well-defined representation space of all the curves. The most important advantage offered by the SRVF framework is that the natural and widely used L^2-measure on $L^2(D, \mathbb{R}^n)$ is invariant to reparameterization. That is, for any two SRVFs q_1 and q_2 and an arbitrarily chosen reparameterization function (nondecreasing) $t = \gamma(\tau)$, we have

$$\|q_1(t) - q_2(t)\|_{L^2} = \|q_1(\gamma(\tau)) - q_2(\gamma(\tau))\|_{L^2}.$$

This property gives us the invariance to some distortions that we mentioned earlier. Furthermore, this reparameterization together with $L^2(D, \mathbb{R}^n)$ forms a quotient manifold. To see this, we introduce some more notations. Let Γ be the set of all diffeomorphisms from $D = [0, 1]$ to $D = [0, 1]$. This set collects all the reparameterization mappings. Γ is a Lie group with composition as the group operation and the identity mapping as the identity element. Then, all the orbits $[q] = \{q \circ \gamma = q(\gamma(t)) \mid \forall \gamma \in \Gamma\}$ together define the quotient manifold $L^2(D, \mathbb{R}^n)/\Gamma$, i.e., all reparameterized $q(t)$ are treated as the same.

Without loss of generality, we assume that all curves are normalized to unitary length, i.e., $\int_D \|\dot{a}(t)\| dt = 1$. The SRVFs

associated with these curves are elements of a unit sphere in the Hilbert space $L^2(D, \mathbb{R}^n)$ because $\int_D \|q(t)\|^2 \, dt = 1$. Therefore, under the curve normalization assumption, instead of $L^2(D, \mathbb{R}^n)$, we consider the following unit sphere manifold:

$$\mathcal{C}^o = \left\{ q \in L^2\big(D, \mathbb{R}^n\big) : \int_D \|q(t)\|^2 \, dt = 1 \right\}.$$

The manifold $\mathcal{C}^o$ has some nice properties (see [37]). For any two points q_0 and q_1 in $\mathcal{C}^o$, the geodesic, i.e., the shortest path connecting two points on the manifold, is given by $\alpha : [0, 1] \to \mathcal{C}^o$

$$\alpha(\tau) = \frac{1}{\sin(\theta)} (\sin(\theta(1-\tau))q_0 + \sin(\theta\tau)q_1) \quad \text{(III.1)}$$

where $\theta = \cos^{-1}(\langle q_0, q_1 \rangle)$ is the length of the geodesic and $\langle \cdot, \cdot \rangle$ is the inner product defined in $L^2(D, \mathbb{R}^n)$. Taking the derivative of α at $\tau = 0$, we obtain the tangent vector at q_0

$$v = \frac{\theta}{\sin(\theta)} [q_1 - \langle q_0, q_1 \rangle q_0]. \quad \text{(III.2)}$$

The above-mentioned formula is regarded as the logarithm mapping $\log_{q_0}(q_1)$ on the manifold $\mathcal{C}^o$, i.e., the map that takes the points on the manifold to the tangent space rooted at some point.

As we are concerned with the shape invariance, i.e., we need to additionally remove the shape-preserving transformations: rotation and curve reparameterization. For this purpose, we introduce rotation group $SO(n)$ for all the rotations in $\mathbb{R}^n$. Combining $SO(n)$ with Γ, we define the equivalence relation between pair of points q_i and q_j, i.e., we claim q_i equivalent to q_j, written as $q_i \sim q_j$, if they are rotated and reparameterized version of each other. The quotient manifold $\mathcal{S}^o = \mathcal{C}^o / (SO(n) \times \Gamma)$ then has the property we need. Each element $[q] \in \mathcal{S}^o$ is an equivalence class defined by

$$[q] = \left\{ Oq(\gamma(t)) \sqrt{\dot{\gamma}(t)} \mid O \in SO(n) \text{ and } \gamma \in \Gamma \right\}.$$

Now, $[q_i]$ is the a family of curves that are transformed version of each other. More concretely, given a set of N unit-length curves $\{a_1(t), \ldots, a_N(t)\}$, their SRVFs are $\{q_1(t), \ldots, q_N(t)\}$ such that $[q_i] \in \mathcal{S}^o$ and $q_i(t)$ is a representative of the equivalent class $[q_i]$. Thus, if $a_i(t)$ is the distorted version of $a_j(t)$, then $[q_i] = [q_j]$, and these two will be reflected as a single point in $\mathcal{S}^o$.

B. Proposed Clustering Method

It is clear that $\mathcal{S}^o$ is a good representation of curves, which is invariant to many distortions that we focus on. From the line of its development, we see that the functional subspace assumption holds because the gradient and rotation are linear operators and reparameterization works on function bases as well. However, the obstacle now is that $\mathcal{S}^o$ is an abstract unitary sphere with no obvious vector form. Recovering the subspace structure is not straightforward. Nonetheless, the tangent space of any smooth manifold at any point is a well-defined vector space with the same dimensionality as the manifold [38]. Observe that the projection to tangent space on unitary sphere $\mathcal{S}^n$ in $\mathbb{R}^{n+1}$ preserves the subspace structure of the points on the sphere. If the foot of a tangent space is from one subspace, then this subspace will have one less dimension in the tangent space. This is illustrated in Fig. 1. This prompts us to use tangent space on $\mathcal{S}^o$. Fortunately, the computation is readily available.

Given any two points $[q_0]$ and $[q_1]$ in $\mathcal{S}^o$, a tangent representative [37] of $[q_1]$ in the tangent space $T_{[q_0]}(\mathcal{S}^o)$ can be calculated in the following way, as suggested in [40] and [41] based on (III.2):

$$\widetilde{v} = \log_{[q_0]}([q_1]) = \frac{\widetilde{\theta}}{\sin(\widetilde{\theta})} [\widetilde{q}_1 - \langle \widetilde{q}_0, \widetilde{q}_1 \rangle \widetilde{q}_0] \quad \text{(III.3)}$$

where $\widetilde{q}$ is the representative of $[q]$ given by the well-defined algorithm in [40] and [41] and $\widetilde{\theta} = \cos^{-1}(\langle \widetilde{q}_0, \widetilde{q}_1 \rangle)$. In fact, $\widetilde{v}$ is the lifting representation of tangent vector $\log_{[q_0]}([q_1])$ in $T_{[q_0]}(\mathcal{S}^o)$ for $[q_1]$, or in other words, the projection of $[q_1]$ in $T_{[q_0]}(\mathcal{S}^o)$.

We are ready to recover the subspace structure now in tangent space. We start with

$$\sum_{j=1}^{N} w_{ij} \log_{[q_i]}\big([q_j]\big) = 0 \quad \forall i = 1, \ldots, N. \quad \text{(III.4)}$$

Note that $\log_{[q_i]}([q_i]) = 0$, i.e., $[q_i]$ is the origin of its tangent space. This introduces indeterminacy for $[q_i]$. Therefore, we have the constraint $\sum_{j=1}^{N} w_{ij} = 1, i = 1, 2, \ldots, N$ to avoid trivial solution and other indeterminacy, such as arbitrary scaling. Moreover, tangent space of every point is utilized as in (III.4) to form a multiple view of the subspaces. The advantage of doing this is not only avoiding a vanishing point but also increasing the stability of subspace recovery process. The next step is to consolidate the multiple views. One critical observation is that although every tangent space is different, depending on the support point, the null space is the same. If $[q_i]$s have subspace structure, the most efficient way to recover null space is to pick points from subspaces consistently, which is reflected as the lowest rank in coefficient matrix $\mathbf{W}$, whose ijth element is w_{ij}. This leads to the following objective:

$$\min_{\mathbf{W}} \lambda \|\mathbf{W}\|_* + \sum_{i=1}^{N} \frac{1}{2} \| \sum_{j=1}^{N} w_{ij} \log_{[q_i]}\big([q_j]\big) \|_{[q_i]}^2,$$
$$\text{s.t.} \sum_{j=1}^{N} w_{ij} = 1, \quad i = 1, 2, \ldots, N. \quad \text{(III.5)}$$

where $\|\cdot\|_{[q]}$ is the metric defined on the manifold, which is defined by the classic L^2 Hilbert metric on the tangent space, quantifying the deviation from 0, and $\|\mathbf{W}\|_*$ is the nuclear norm as a convex approximation to matrix rank. Note that the above-mentioned objective function reflects the tradeoff between low rank and null space recovery, i.e., (III.4), controlled by λ.

Denote $\mathbf{w}_i$ the ith row of matrix $\mathbf{W}$ and define

$$B_{jk}^i = \langle \log_{[q_i]}\big([q_j]\big), \log_{[q_i]}([q_k]) \rangle. \quad \text{(III.6)}$$

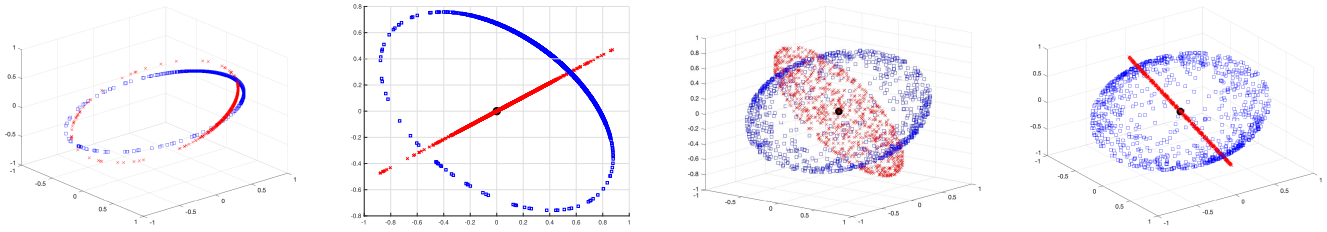


Fig. 1. Illustration of projection to tangent space. Different symbols are for different subspaces on the sphere. The black dots in (b) and (d) are the origin of the tangent spaces. (a) and (b) Original points on $\mathcal{S}^2$ and their projections to a tangent space footed on one of the points. (c) and (d) Two 2-D subspaces in $\mathcal{S}^3$. (c) One view of the projection. (d) Rotation of (c) to reveal the reduced subspace.

Then, with some algebraic manipulation, we can rewrite (III.5) into the following simplified form:

$$\min_{\mathbf{W}} \lambda\|\mathbf{W}\|_* + \sum_{i=1}^{N} \mathbf{w}_i \mathbf{B}^i \mathbf{w}_i^T, \quad \text{s.t.} \sum_{j=1}^{N} w_{ij} = 1, \quad i = 1, 2, \ldots, N \tag{III.7}$$

where $\mathbf{B}^i = (B^i_{jk})$.

Once solved, $\mathbf{W}$ can be used as an affinity to build a graph. As $\mathbf{W}$ has a consistent pattern within each subspace, the graph should consist of several connected components corresponding to subspaces and graph cut can then be applied to segment them.

IV. Optimization

A. Algorithm

To solve the objective function in (III.7), we use the LADMAP [41], [42]. First, take the augmented Lagrangian of the objective (III.7)

$$L = \lambda\|\mathbf{W}\|_* + \frac{1}{2}\sum_{i=1}^{N} \mathbf{w}_i \mathbf{B}^i \mathbf{w}_i^T + \langle \mathbf{y}, \mathbf{W}\mathbf{1} - \mathbf{1}\rangle + \frac{\beta}{2}\|\mathbf{W}\mathbf{1} - \mathbf{1}\|_F^2 \tag{IV.1}$$

where $\mathbf{y}$ is the Lagrangian multiplier (vector) corresponding to the equality constraint $\mathbf{W}\mathbf{1} = \mathbf{1}$, $\|\cdot\|_F$ is the matrix Frobenius-norm, and β is the proximal parameter that will be updated in the iterative algorithm to be introduced.

Denote by $F(\mathbf{W})$ the function defined by (IV.1) except for the first term $\lambda\|\mathbf{W}\|_*$. We adopt a linearization of $F(\mathbf{W})$ at the current location $\mathbf{W}^{(k)}$ in the iteration process, that is, we approximate $F(\mathbf{W})$ by the following linearization with a proximal term:

$$F(\mathbf{W}) \approx F\big(\mathbf{W}^{(k)}\big) + \langle \partial F\big(\mathbf{W}^{(k)}\big), \mathbf{W} - \mathbf{W}^{(k)}\rangle + \frac{\eta_W \beta_k}{2}\|\mathbf{W} - \mathbf{W}^{(k)}\|_F^2$$

where η_W is an approximate constant with a suggested value given by $\eta_W = \max\{\|B_i\|^2\} + N + 1$ (this value is presented in the convergence theorem in Section IV-C), and $\partial F(\mathbf{W}^{(k)})$ is the gradient matrix of $F(\mathbf{W})$ at $\mathbf{W}^{(k)}$. Denote by $\mathbf{B}$ the 3-order tensor whose ith frontal slice is $\mathbf{B}^i$. Write $\mathbf{W} \odot \mathbf{B}$ the matrix whose ith row is given by $\mathbf{w}_i \mathbf{B}^i$. Then, it is easy to show

$$\partial F\big(\mathbf{W}^{(k)}\big) = \mathbf{W} \odot \mathbf{B} + \mathbf{y}\mathbf{1}^T + \beta_k(\mathbf{W}\mathbf{1} - \mathbf{1})\mathbf{1}^T. \tag{IV.2}$$

Then, (IV.1) can be approximated by linearization, and $\mathbf{W}$ will be updated by the following:

$$\mathbf{W}^{(k+1)} = \arg\min_{\mathbf{W}} \lambda\|\mathbf{W}\|_* + \frac{\eta_W \beta_k}{2}\left\|\mathbf{W} - \left(\mathbf{W}^{(k)} - \frac{1}{\eta_W \beta_k}\partial F\big(\mathbf{W}^{(k)}\big)\right)\right\|_F^2. \tag{IV.3}$$

Problem (IV.3) admits a closed-form solution by using SVD thresholding operator [43], given by

$$\mathbf{W}^{(k+1)} = U_W S_{\frac{\lambda}{\eta_W \beta_k}}(\Sigma_W) V_W^T \tag{IV.4}$$

where $U_W \Sigma_W V_W^T$ is the SVD of $\mathbf{W}^{(k)} - \frac{1}{\eta_W \beta_k}\partial F(\mathbf{W}^{(k)})$ and $S_\tau(\cdot)$ is the singular value thresholding (SVT) [43], [44] operator defined by

$$S_\tau(\Sigma) = \operatorname{diag}(\max\{|\Sigma_{ii}| - \tau, 0\}). \tag{IV.5}$$

The updating rule for $\mathbf{y}$ is

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \beta_k\big(\mathbf{W}^{(k)}\mathbf{1} - \mathbf{1}\big) \tag{IV.6}$$

and the updating rule for β_k

$$\beta_{k+1} = \min\{\beta_{\max}, \rho\beta_k\} \tag{IV.7}$$

where

$$\rho = \begin{cases} \rho_0, & \text{if } \beta_k\|\mathbf{W}^{k+1} - \mathbf{W}^k\| \le \varepsilon_1, \\ 1, & \text{otherwise.} \end{cases}$$

We summarize the above as Algorithm 1. Once the coefficient matrix $\mathbf{W}$ is found, spectral clustering is applied to the affinity matrix $\frac{|\mathbf{W}|+|\mathbf{W}|^T}{2}$ to obtain the segmentation of the data. In particular, we use nCUT [33] for its good performance in both accuracy and efficiency in spectral clustering. We call our method rFMC.

B. Complexity Analysis

For ease of analysis, we first define some symbols used in the following. Let K and r denote the total number of iterations and the lowest rank of the matrix $\mathbf{W}$, respectively. The size of $\mathbf{W}$ is $N \times N$. The major computation cost of our proposed method contains two parts, calculating all $\mathbf{B}^i$'s and

Algorithm 1 Optimization for (III.7)

Require: $\{\mathbf{X}_i\}_{i=1}^N$, λ

1: Initialize: $\mathbf{W} = \mathbf{0}$, $\mathbf{y} = \mathbf{0}$, $\beta = 0.1$, $\beta_{\max} = 10$, $\rho^0 = 1.1$, $\eta = \max\{\|\mathbf{B}^i\|_F\} + N + 1$, $\epsilon_1 = 1e^{-4}$, $\epsilon_2 = 1e^{-4}$

2: Construct each $\mathbf{B}^i$ as per (III.6)

3: **while** not converged **do**

4: Update $\mathbf{W}$ using (IV.4)

5: Check convergence criteria

$$\beta^{(k)}\|\mathbf{W}^{(k+1)} - \mathbf{W}^{(k)}\|_F \le \epsilon_1$$
$$\|\mathbf{W}\mathbf{1} - \mathbf{1}\|_F \le \epsilon_2$$

6: Update Lagrangian Multiplier

$$\mathbf{y}^{(k+1)} = \mathbf{y}^k + \beta^{(k)}(\mathbf{W}\mathbf{1} - \mathbf{1})^T$$

7: Update ρ

$$\rho = \begin{cases} \rho_0 & \text{if } \beta^{(k)}\|\mathbf{W}^{(k+1)} - \mathbf{W}^{(k)}\|_F \le \epsilon_1 \\ 1 & \text{otherwise,} \end{cases}$$

8: Update β

$$\beta^{(k+1)} = \min\left(\beta_{\max}, \rho\beta^{(k)}\right)$$

9: **end while**

return W

updating $\mathbf{W}$. In terms of the formula (III.6) through (III.2) and (III.3), the computational complexity of Log algorithm is $O(T^2)$ where T is the number of terms in a discretized curves; therefore, the complexity of B^i_{jk} is at most $O(T^2)$ and $\mathbf{B}^i$'s computational complexity is $O(N^2T^2)$. Thus, the total for all the $\mathbf{B}^i$ is $O(N^3T^2)$. In each iteration of the algorithm, the singular value thresholding is adopted to update the low-rank matrix $\mathbf{W}$ whose complexity is $O(rN^2)$ [32]. Suppose that the algorithm is terminated after K iterations, and the overall computational complexity is given by

$$O\left(N^3T^2\right) + O\left(KrN^2\right).$$

C. Convergence Analysis

Algorithm 1 is adopted from the algorithm proposed in [42]. However due to the terms of $\mathbf{B}^i$'s in the objective function (IV.1), the convergence theorem proved in [42] cannot be directly applied to this case as the linearization is implemented on both the augmented Lagrangian terms and the term involving $\mathbf{B}^i$'s. Fortunately, we can employ the revised approach, presented in [45], to prove the convergence for the algorithm. Without repeating all the details, we present the convergence theorem for Algorithm 1 as follows.

Theorem 1 (Convergence of Algorithm 1): If $\eta_W \ge \max\{\|B_i\|^2\} + N + 1$, $\sum_{k=1}^{+\infty}\beta_k^{-1} = +\infty$, $\beta_{k+1} - \beta_k > C_0\dfrac{\sum_i \|B_i\|^2}{\eta_W - \max\{\|B_i\|^2\} - N}$, where C_0 is a given constant and $\|\cdot\|$ is the matrix spectral norm; then, the sequence $\{W^k\}$ generated by Algorithm 1 converges to an optimal solution to problem (III.7).

In all the experiments, we have conducted that the algorithm converges very fast at $K < 100$.

V. Experimental Analysis

In this section, we evaluate the clustering performance of rFMC on synthetic, semisynthetic, and real-world data sets. We compare our algorithm with two baseline algorithms, k-means and spectral clustering of a DTW distance matrix, and the state-of-the-art Bayesian clustering of curves. We also compare against LRR and SSC, two highly cited multivariate subspace clustering methods. Note that we used nCUT to find final clustering solution for LRR and SSC as well.

In an effort to maximize transparency and repeatability, all code and data used for these experiments can be found online at https://staff.scem.uws.edu.au/~yiguo/code/rfmc.

To help evaluate consistency, we fixed the parameters to the same values for every experiment. Parameters were selected by testing a wide range of values over all data sets so that the best average result for each method was obtained. For rFMC and LRR, λ was set to 0.1, and for SSC, λ was 0.05. Overall, we found that the segmentation accuracy of LRR and SSC did not vary that much with changes in λ. For our DTW baseline algorithm, we set the warping window to 10% of the data length, which has been shown to be suitable in most cases [22].

Segmentation accuracy was measured using the subspace clustering accuracy (SCA) metric [31], which is defined as

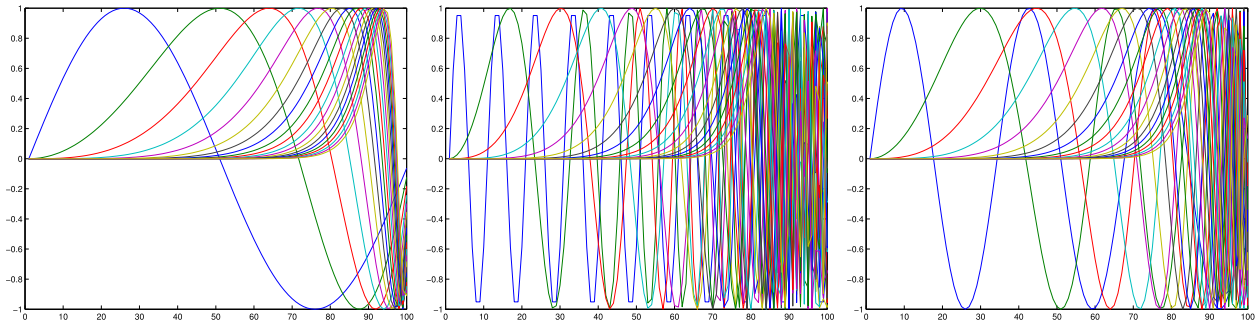
$$\text{SCA} = 1 - \frac{\text{num. of misclassified points}}{\text{total num. of points}} \tag{V.1}$$

where higher SCA means greater clustering accuracy. The SCA metric is taken over all possible pairwise assignments of clusters.

A. Toy Synthetic Data Clustering

First, we attempt to verify that rFMC achieves its design purpose for robustness against distortions in subspace recovery, while other methods without the consideration of the special properties of functions, such as LRR and k-means, would be inferior. In this test, 2–8 clusters were created consisting of twenty 1-D curves of length 100. The curves in each cluster were sine waves, with each cluster corresponding to a unique frequency from 0.1 to 40 Hz. Within each cluster, we applied progressive amounts of warping. See Fig. 2 for an example of data from three synthetically generated clusters. For each number of clusters setting, we repeated the experiment 50 times with new data generated each time to obtain basic statistics of SAC.

Results are reported using subspace clustering accuracy and can be found in Table I and Fig. 3. Note that in Table I and the following ones, the number besides rFMC indicates the number of clusters tested, and $\bar{T}$ in the tables is the average run time for the algorithms in the tests. This is a challenging data set due to a large number of distortions. However, in this experiment, rFMC achieves very high clustering accuracy in terms of the statistics of accuracy values. From Fig. 3, it is clearly seen that rFMC outperforms other



(a) (b) (c)

Fig. 2. Examples of clusters and their curves generated in the synthetic data experiment. Each cluster has a base sine curve (the leftmost blue curve) that is progressively warped with each successive instantiation. (a) Cluster 1. (b) Cluster 2. (c) Cluster 3.

TABLE I
SYNTHETIC DATA RESULTS WITH THREE, FIVE, AND EIGHT CLUSTERS

	Mean	Median	Max	Min	Std	$\bar{T}$
kmeans	44.1%	41.7%	60.0%	35.0%	7.4%	0.01
DTW	58.9%	59.2%	81.7%	36.7%	12.0%	0.11
LRR	69.5%	68.3%	91.7%	36.7%	14.7%	0.04
SSC	59.6%	51.7%	93.3%	38.3%	15.6%	0.10
Bayesian	62.9%	65.0%	100.0%	25.0%	23.7%	48.71
rFMC(3)	**80.5%**	**83.3%**	**98.3%**	**36.7%**	17.7%	11.18
kmeans	35.1%	35.0%	48.0%	23.0%	6.4%	0.01
DTW	47.3%	48.0%	61.0%	24.0%	9.0%	0.30
LRR	56.9%	59.0%	83.0%	**27.0%**	13.9%	0.14
SSC	46.0%	42.0%	79.0%	25.0%	12.3%	0.11
Bayesian	51.8%	55.5%	78.0%	19.0%	14.7%	137.96
rFMC(5)	**63.9%**	**64.5%**	**96.0%**	**27.0%**	15.6%	27.55
kmeans	28.0%	27.5%	39.4%	20.0%	4.1%	0.01
DTW	40.0%	40.0%	48.8%	24.4%	5.2%	0.65
LRR	48.6%	48.8%	65.0%	24.4%	8.0%	0.27
SSC	38.2%	40.6%	56.9%	24.4%	6.6%	0.15
Bayesian	42.7%	42.8%	**69.4%**	21.3%	10.3%	411.06
rFMC(8)	**49.6%**	**49.4%**	66.2%	**31.2%**	7.2%	64.73

methods in terms of SAC mean and median. When the number of clusters grows, the performance decreases due to increasing difficulty. Note that when the number of clusters reaches 8, the clustering accuracies of all methods drop below 50%, which is not informative. Moreover, some methods, such as the Bayesian become unstable, for some reason. Thus, we set 8 as the maximum number of clusters in all tests. In the following experiments, we test 3, 5, and 8 clusters to observe the behavior of different methods in terms of both clustering accuracy and the trends when the number of clusters varies.

B. Semisynthetic Thermal Infrared Spectra Clustering

We assemble semisynthetic data from a library of pure infrared hyperspectral mineral data [29]. For each cluster, we pick one spectral sample from the library as a basis. Each curve basis is then randomly shifted and stretched in a random portion. This random warping is performed 20 times to produce the curves for each cluster. See Fig. 4 for an example of data used in this experiment. Again, as in the previous experiment, we repeated the test 50 times.

Results are reported in Table II. The results show that nonfunctional-based methods cannot accurately cluster data

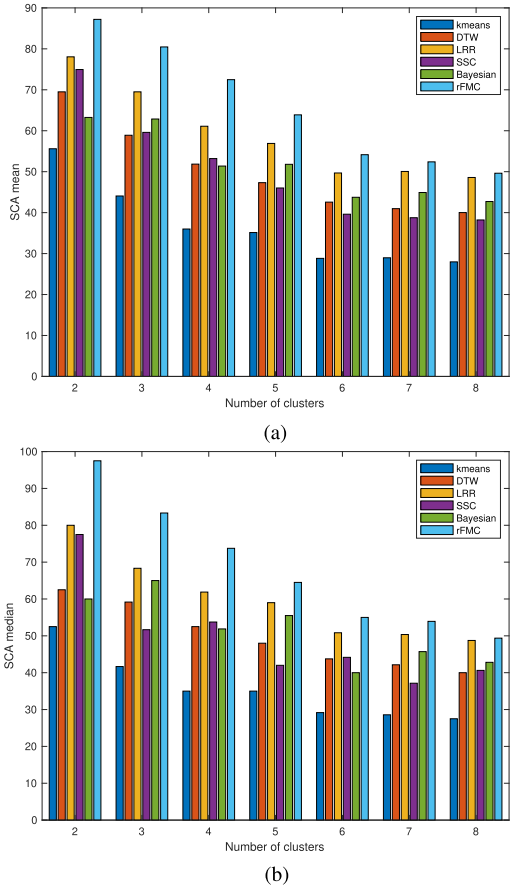


Fig. 3. SCA means and medians of different methods on synthetic data. (a) SCA means. (b) SCA medians.

with this sort of structural contamination, which is commonly found in this type of data due to impurities in the mineral samples. On the other hand, rFMC almost perfectly clustered the data with superb consistency across all cases. The closest competitor was the Bayesian method, which also performed well by clustering accurately most of the time at the cost of extremely high computation load. However, in some cases, the clusters produced were of poor quality, which can be observed in the minimum accuracy and standard deviation statistics. Therefore, rFMC is far more reliable and efficient at clustering this data than other methods.

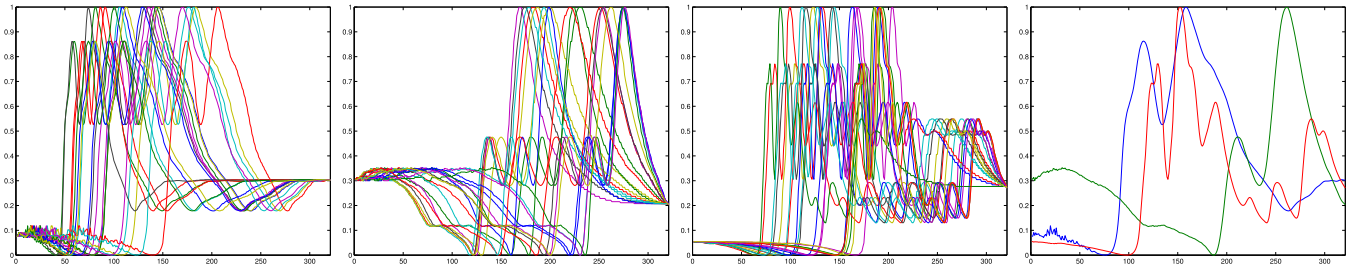


Fig. 4. Example plots of curves used in the semisynthetic TIR data experiment. Each cluster has a base curve from the TIR library. The curves for each cluster have been shifted and stretched randomly from the base. (a) Cluster 1. (b) Cluster 2. (c) Cluster 3. (d) Base curves.

TABLE II
THERMAL INFRARED RESULTS

	Mean	Median	Max	Min	Std	$\bar{T}$
kmeans	58.6%	58.3%	73.3%	43.3%	6.1%	0.01
DTW	65.8%	65.8%	81.7%	46.7%	6.0%	0.74
LRR	60.2%	60.8%	66.7%	43.3%	5.0%	0.14
SSC	48.1%	46.7%	61.7%	40.0%	4.8%	0.13
Bayesian	94.7%	**100.0%**	**100.0%**	66.7%	12.3%	311.66
rFMC(3)	**100.0%**	**100.0%**	**100.0%**	**100.0%**	0.0%	190.50
kmeans	35.2%	35.0%	43.0%	29.0%	3.2%	0.01
DTW	50.3%	50.0%	63.0%	37.0%	5.2%	2.36
LRR	33.3%	33.0%	39.0%	27.0%	2.8%	0.26
SSC	37.1%	37.0%	46.0%	31.0%	3.4%	0.27
Bayesian	96.0%	**100.0%**	**100.0%**	80.0%	8.1%	1337.20
rFMC(5)	**100.0%**	**100.0%**	**100.0%**	**100.0%**	0.0%	278.18
kmeans	39.4%	38.7%	46.2%	34.4%	2.5%	0.01
DTW	45.8%	45.6%	55.0%	38.1%	4.1%	5.64
LRR	27.0%	26.9%	32.5%	23.1%	2.1%	0.53
SSC	32.3%	32.5%	38.1%	26.2%	2.7%	0.72
Bayesian	81.9%	87.5%	87.5%	69.4%	6.6%	3246.81
rFMC(8)	**92.8%**	**96.9%**	**100.0%**	**78.8%**	7.0%	658.50

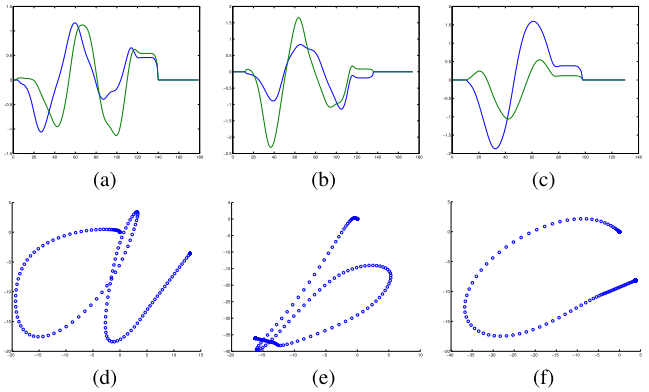


Fig. 5. Example data from the character velocity data set. The top row plots the x and y pen tip velocities over time for three sample characters. The bottom row shows the corresponding character reconstruction by integrating the pen tip velocity data (for visualization only). (a) Trajectories for "a." (b) Trajectories for "b." (c) Trajectories for "c." (d) Reconstructed "a." (e) Reconstructed "b." (f) Reconstructed "c."

C. Handwriting Character Velocities

In this experiment, a real-world data set consisting of a collection of pen tip trajectories of handwritten English characters was used to evaluate performance. The data set consists of pen position data collected by a digitization tablet at 200 Hz, which is then converted to horizontal and vertical velocities [46], [47]. These 2-D trajectory curves are normalized such that the mean of each curve is close to zero. See Fig. 5 for some examples of this data. Fig. 6 shows the example plots of curves used in this experiment.

For each run of this test, 20 characters were randomly selected from three, five, and eight random character classes. The data as originally released have been carefully produced and processed so that trajectories for each character are extremely similar, far more so than is realistic. For example, the start time for each character has been aligned; furthermore, the writing speed, character size, and variance in velocity over time are extremely consistent. Therefore, to make the data more realistic, we randomly globally shift each character so that their start times vary. Furthermore, we randomly globally stretch and shrink each trajectory to account for different writing speeds, we also scale the trajectories by applying constant factors to account for character size, and finally, we perform local warping (as done in the semisynthetic experiment) to account for variance in speed over time.

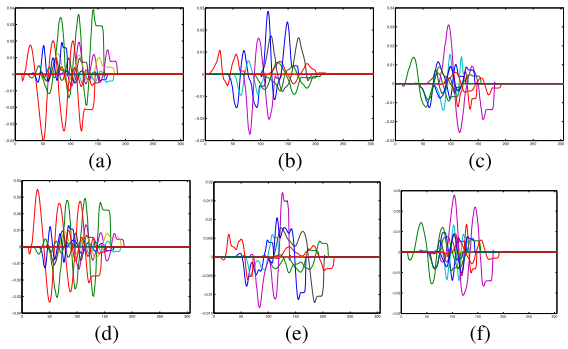


Fig. 6. Example plots of curves used in the character velocity experiment. Each cluster consists of randomly selected characters from each class that are then subject to a combination of shifting, warping, stretching or shrinking, and scaling. The top row shows the curves from the pen tip velocity in the X-direction over time, and the bottom row shows the same but for the Y-direction. (a) Cluster 1-X. (b) Cluster 2-X. (c) Cluster 3-X. (d) Cluster 1-Y. (e) Cluster 2-Y. (f) Cluster 3-Y.

Results can be found in Table III. RFMC shows excellent performance with a median accuracy of 86% for the case of three classes on this extremely challenging data set. The closest competitors only reach a median clustering accuracy of 50%. A similar comparison can be observed in other cases. It is clear to see that rFMC outperforms other methods in all metrics in all cases of various numbers of classes.

TABLE III
CHARACTER VELOCITY RESULTS

	Mean	Median	Max	Min	Std	$\bar{T}$
kmeans	0.0%	0.0%	0.0%	0.0%	0.0%	0.00
DTW	48.2%	46.7%	63.3%	40.0%	6.4%	0.18
LRR	48.3%	46.7%	70.0%	36.7%	7.7%	0.01
SSC	48.8%	50.0%	70.0%	36.7%	6.7%	0.05
Bayesian	47.5%	33.3%	83.3%	33.3%	16.3%	79.19
rFMC(3)	**84.6%**	**86.7%**	**100.0%**	**53.3%**	14.4%	46.38
kmeans	0.0%	0.0%	0.0%	0.0%	0.0%	0.00
DTW	37.7%	37.0%	50.0%	30.0%	4.0%	0.48
LRR	37.4%	36.0%	48.0%	28.0%	4.9%	0.03
SSC	37.4%	38.0%	46.0%	30.0%	4.0%	0.09
Bayesian	54.1%	52.0%	78.0%	0.0%	13.1%	291.51
rFMC(5)	**67.9%**	**68.0%**	**90.0%**	**50.0%**	9.9%	70.10
kmeans	0.0%	0.0%	0.0%	0.0%	0.0%	0.00
DTW	33.4%	33.8%	41.2%	27.5%	3.7%	1.14
LRR	32.7%	32.5%	40.0%	27.5%	3.0%	0.07
SSC	30.8%	30.0%	37.5%	26.2%	2.7%	0.15
Bayesian	48.4%	52.5%	65.0%	0.0%	15.6%	725.15
rFMC(8)	**61.3%**	**60.0%**	**77.5%**	**43.8%**	7.7%	163.35

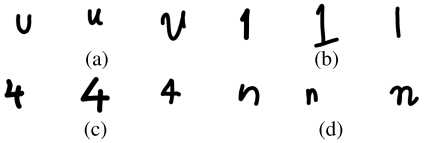


Fig. 7. Example data from the character classification data set. Note that the writing style and typeface varies within each character class. There is also variance in size and position of the characters. (a) "u" character. (b) "1" character. (c) "4" character. (d) "n" character.

TABLE IV
CHARACTER TRAJECTORY RESULTS

	Mean	Median	Max	Min	Std	$\bar{T}$
kmeans	55.6%	55.0%	80.0%	36.7%	9.8%	0.01
DTW	58.4%	58.3%	85.0%	38.3%	10.2%	0.91
LRR	53.7%	53.3%	73.3%	36.7%	8.9%	0.25
SSC	52.3%	51.7%	75.0%	38.3%	9.2%	0.22
Bayesian	14.6%	0.0%	98.3%	0.0%	32.5%	698.81
rFMC(3)	**82.3%**	**88.3%**	**100.0%**	**43.3%**	15.2%	155.60
kmeans	46.7%	46.0%	59.0%	36.0%	5.1%	0.01
DTW	48.9%	48.5%	65.0%	34.0%	8.3%	2.31
SSC	27.8%	28.0%	30.0%	26.0%	1.2%	1.09
Bayesian	2.8%	0.0%	80.0%	0.0%	13.8%	1657.19
rFMC(5)	**70.2%**	**70.0%**	**95.0%**	**53.0%**	8.8%	386.84
kmeans	27.3%	26.9%	38.1%	22.5%	2.9%	0.01
DTW	39.4%	39.4%	49.4%	29.4%	4.4%	7.17
SSC	22.1%	21.9%	25.0%	20.0%	1.1%	2.71
Bayesian	0.0%	0.0%	0.0%	0.0%	0.0%	4589.25
rFMC(8)	**59.1%**	**60.0%**	**70.6%**	**43.8%**	7.7%	1277.38

D. Handwriting Character Trajectories

In this experiment, we used the Chars74K [48] data set, which consists of pen tip positions (different from trajectories) of handwritten English characters. The data set consists of 62 character classes with 55 samples per class. Different from the previous English character data set, the writing style or the typeface varies within each character class. For example, some people choose to close the top of the character "4," while others leave it open or they choose to write their characters with differing amounts of serifs as can be seen with the character "1." See Fig. 7 for visual examples. This creates a significant problem for clustering as the shape of the data within a class varies significantly, and it is actually best to treat these different font faces as separate classes for greater accuracy. However, we do not divide the classes into separate font faces.

Similar to the previous experiment for each run of this test, 20 characters were randomly selected from three, five, and eight random character classes, and 50 runs were performed. However, different from the previous experiment, we do not apply any further postprocessing to reduce alignment since this data set is relatively unprocessed. Results can be found in Table IV. In spite of the aforementioned challenges with this data set, rFMC leads by a significant margin. Notably, the Bayesian clustering suffered many clustering failures leading to very poor accuracy. This was due to implementation limitations in the original code provided from [28]. We also have to exclude LRR for the cases where the number of classes is more than 3 because it failed on most random samples.

TABLE V
SIGN LANGUAGE RESULTS

	Mean	Median	Max	Min	Std	$\bar{T}$
kmeans	64.4%	61.7%	95.1%	40.7%	14.8%	0.02
DTW	63.8%	63.0%	92.6%	51.9%	7.8%	0.46
LRR	67.3%	66.0%	96.3%	42.0%	14.3%	0.27
SSC	56.2%	54.9%	84.0%	38.3%	10.2%	0.43
Bayesian	96.7%	**100.0%**	**100.0%**	64.2%	9.4%	186.10
rFMC(3)	**98.3%**	**100.0%**	**100.0%**	**67.9%**	5.4%	98.88
kmeans	51.7%	51.1%	79.3%	31.9%	10.5%	0.01
DTW	62.2%	62.6%	79.3%	45.2%	9.0%	1.25
LRR	59.1%	59.6%	89.6%	34.1%	10.6%	0.38
SSC	45.4%	45.2%	75.6%	31.1%	9.0%	0.99
Bayesian	90.0%	**99.3%**	**100.0%**	**70.4%**	11.6%	628.98
rFMC(5)	**91.7%**	97.0%	**100.0%**	63.0%	10.0%	121.79
kmeans	43.9%	44.2%	67.1%	26.9%	7.9%	0.01
DTW	56.0%	56.0%	70.4%	45.4%	5.6%	2.52
LRR	48.7%	48.6%	63.0%	29.6%	6.5%	0.56
SSC	39.8%	40.0%	54.6%	25.9%	6.4%	6.72
Bayesian	**87.0%**	84.7%	**100.0%**	69.9%	9.4%	1438.01
rFMC(8)	85.1%	**86.1%**	95.8%	**70.4%**	7.3%	313.89

E. Sign Language Word Clustering

In this final experiment, we use the Australian Sign Language (Auslan) Signs (high quality) data set [49]. This data set consists of a single native Auslan signer performing 91 different signs (the classes) with 27 samples per sign. The signer wore motion capture gloves that captured the position (x, y, z), roll, pitch, and yaw of each hand along with finger bend measurements. This data was captured at 100 Hz, and no postprocessing was applied. The signs were collected three at a time over a period of nine weeks, so there is noticeable variation within each class.

As with the previous experiments, for each run, 20 data points (signed words) were randomly selected from three, five, and eight random word classes, and 50 runs were performed. Only the position, roll, pitch, and yaw channels were used since the finger bend measurements were far too noisy and unreliable to be of use. We also performed a parallel test to determine the effectiveness of smoothing as the first attempt for noise handling. A multichannel total variation-based smoothing method was used [50]. Results can be found in Tables V and VI. Overall, rFMC performed

TABLE VI
SMOOTHED SIGN LANGUAGE RESULTS

	Mean	Median	Max	Min	Std	$\bar{T}$
kmeans	58.0%	56.8%	100.0%	0.0%	19.6%	0.02
DTW	63.3%	63.0%	92.6%	46.9%	7.8%	0.46
LRR	63.0%	62.3%	90.1%	39.5%	12.5%	0.27
SSC	56.3%	55.6%	84.0%	38.3%	10.2%	0.43
Bayesian	95.0%	**100.0%**	**100.0%**	63.0%	11.1%	186.64
rFMC(3)	**97.7%**	**100.0%**	**100.0%**	**69.1%**	5.6%	98.83
kmeans	52.5%	51.1%	31.1%	82.2%	10.3%	0.01
DTW	61.0%	59.3%	42.2%	79.3%	9.3%	1.26
LRR	53.8%	51.9%	33.3%	85.9%	10.0%	0.38
SSC	46.0%	47.0%	31.1%	67.4%	9.1%	0.99
Bayesian	**90.6%**	**98.9%**	**100.0%**	54.8%	11.6%	629.30
rFMC(5)	89.2%	94.8%	**100.0%**	**62.2%**	11.6%	121.06
kmeans	43.4%	42.6%	27.3%	60.2%	6.3%	0.01
DTW	55.4%	57.4%	40.3%	64.8%	5.4%	2.48
LRR	44.1%	44.2%	32.4%	59.3%	5.5%	0.58
SSC	40.1%	40.5%	26.4%	53.2%	6.1%	6.48
Bayesian	**86.3%**	**85.9%**	**100.0%**	54.2%	11.0%	1432.16
rFMC(8)	80.9%	82.4%	90.3%	**67.1%**	6.6%	301.43

slightly better than or similar to the Bayesian clustering and significantly better than the baseline methods. We noticed that smoothing the data actually decreased the clustering performance of all methods, which indicates the nontriviality of noise removal.

VI. CONCLUSION

In this article, we proposed an algorithm called rFMC to reliably and accurately cluster functional data in terms of their subspaces. This is a highly challenging problem as functional data from the same class can vary greatly due to stretching, shrinking, nonuniformly warping, and scaling. We achieved this by representing functional data in curves manifold invariant to these distortions and addressed the challenge of lack of vector forms. The analysis is performed in tangent spaces of each point in the manifold via the recovery of null spaces of the log maps of the points to solve the multiple view problems. An optimization scheme with a convergence guarantee was also provided to realize the model.

Nevertheless, this article still leaves many areas open for further research. First, we only address the data on the manifold of open curves; however, much of the data in recognition and computer vision tasks will lie on the manifold of closed curves. Moreover, our focus was on robustness against geometric distortions. Additive noise will be carried through the transformations onto the curve manifold and affect the performance. However, the mechanism is not fully understood although the proposed algorithm can be applied in this case. Nonetheless, it is interesting to study the noise to further improve the results. We leave solving these problems for future research.

REFERENCES

[1] C. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). Berlin, Germany: Springer-Verlag, 2006.

[2] E. A. Maharaj, "Cluster of time series," *J. Classification*, vol. 17, no. 2, pp. 297–314, 2000.

[3] K. Kalpakis, D. Gada, and V. Puttagunta, "Distance measures for effective clustering of ARIMA time-series," in *Proc. IEEE Int. Conf. Data Mining*, Nov.Dec. 2001, pp. 273–280.

[4] D.-J. Lee, R. B. Schoenberger, D. Shiozawa, X. Xu, and P. Zhan, "Contour matching for a fish recognition and migration-monitoring system," *Proc. SPIE*, vol. 5606, pp. 37–48, Dec. 2004.

[5] S. J. Gaffney, "Probabilistic curve-aligned clustering and prediction with regression mixture models," Ph.D. dissertation, School Inf. Comput. Sci., Univ. California, Irvine, CA, USA, 2004.

[6] T. Rakthanmanon *et al.*, "Searching and mining trillions of time series subsequences under dynamic time warping," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2012, pp. 262–270.

[7] M. Müller, *Information Retrieval for Music and Motion*. Berlin, Germany: Springer, 2007, pp. 211–226.

[8] F. Ferraty and Y. Romain, Eds., *The Oxford Handbook of Functional Data Analysis*. New York, NY, USA: Oxford Univ. Press, 2011.

[9] A. Srivastava, E. Klassen, S. H. Joshi, and I. H. Jermyn, "Shape analysis of elastic curves in Euclidean spaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 7, pp. 1415–1428, Jul. 2011.

[10] A. Srivastava, W. Wu, S. Kurtek, E. Klassen, and J. S. Marron, "Registration of functional data using Fisher-rao metric," 2011, *arXiv:1103.3817*. [Online]. Available: http://arxiv.org/abs/1103.3817

[11] M. T. Bahadori, D. Kale, Y. Fan, and Y. Liu, "Functional subspace clustering with application to time series," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 228–237.

[12] F. Petitjean, G. Forestier, G. I. Webb, A. E. Nicholson, Y. Chen, and E. Keogh, "Dynamic time warping averaging of time series allows faster and more accurate classification," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2014, pp. 470–479.

[13] J. Ramsay and B. W. Silverman, *Functional Data Analysis* (Springer Series in Statistics). Springer, 2005.

[14] J. Jacques and C. Preda, "Functional data clustering: A survey," *Adv. Data Anal. Classification*, vol. 8, no. 3, pp. 231–255, Sep. 2014.

[15] T. Rakthanmanon, "Addressing big data time series: Mining trillions of time series subsequences under dynamic time warping," *ACM Trans. Knowl. Discovery Data*, vol. 7, no. 3, pp. 1–31, 2013.

[16] J. D. Tucker, W. Wu, and A. Srivastava, "Generative models for functional data using phase and amplitude separation," *Comput. Statist. Data Anal.*, vol. 61, pp. 50–66, May 2013.

[17] J. Su, A. Srivastava, and F. W. Huffer, "Detection, classification and estimation of individual shapes in 2D and 3D point clouds," *Comput. Statist. Data Anal.*, vol. 58, pp. 227–241, Feb. 2013.

[18] S. H. Joshi, E. Klassen, A. Srivastava, and I. Jermyn, "A novel representation for riemannian analysis of elastic curves in R^n," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–7.

[19] T. W. Liao, "Clustering of time series data—A survey," *Pattern Recognit.*, vol. 38, no. 11, pp. 1857–1874, 2005.

[20] E. J. Keogh and M. J. Pazzani, "Scaling up dynamic time warping to massive datasets," in *Principles of Data Mining and Knowledge Discovery*. Dublin, Ireland: Springer, 1999, pp. 1–11.

[21] Y.-S. Jeong, M. K. Jeong, and O. A. Omitaomu, "Weighted dynamic time warping for time series classification," *Pattern Recognit.*, vol. 44, no. 9, pp. 2231–2240, Sep. 2011.

[22] C. A. Ratanamahatana and E. Keogh, "Everything you know about dynamic time warping is wrong," in *Proc. 3rd Workshop Mining Temporal Sequential Data*, 2004, pp. 1–152.

[23] T. Oates, L. Firoiu, and P. R. Cohen, "Clustering time series with hidden Markov models and dynamic time warping," in *Proc. Workshop Neural, Symbolic Reinforcement Learn. Methods Sequence Learn. (IJCAI)*, 1999, pp. 17–21.

[24] S. J. Gaffney and P. Smyth, "Joint probabilistic curve clustering and alignment," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 473–480.

[25] J. D. Banfield and A. E. Raftery, "Model-based Gaussian and non-Gaussian clustering," *Biometrics*, vol. 49, no. 3, pp. 803–821, Sep. 1993.

[26] P. McCullagh, "Marginal likelihood for distance matrices," *Stat. Sinica*, vol. 19, no. 2, pp. 631–649, Apr. 2009.

[27] Z. Zhang, Q. Xie, and A. Srivastava, "Elastic registration and shape analysis of functional objects," *Geometry Driven Statist.*, vol. 121, pp. 218–238, Sep. 2015.

[28] Z. Zhang, D. Pati, and A. Srivastava, "Bayesian clustering of shapes of curves," *J. Stat. Planning Inference*, vol. 166, pp. 171–186, Nov. 2015.

[29] Y. Guo, J. Gao, and F. Li, "Spatial subspace clustering for hyperspectral data segmentation," in *Proc. 3rd Int. Conf. Digit. Inf. Process. Commun.*, 2013, pp. 180–190.

[30] Y. Guo, J. Gao, and F. Li, "Random spatial subspace clustering," *Knowl.-Based Syst.*, vol. 74, pp. 106–118, Jan. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950705114003980
[31] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.
[32] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.
[33] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[34] Y. Guo and M. Berman, "A comparison between subset selection and l1 regularisation with an application in spectroscopy," *Chemometric Intell. Lab. Syst.*, vol. 118, pp. 127–138, Aug. 2012.
[35] M. Spivak, *A Comprehensive Introduction to Differential Geometry*. Houston, TX, USA: INC, 1999.
[36] D. Robinson, "Functional analysis and partial matching in the square root velocity framework," Ph.D. dissertation, Dept. Math., Florida State Univ., Tallahassee, FL, USA, 2012.
[37] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ, USA: Princeton Univ. Press, 2008.
[38] J. M. Lee, *Introduction to Smooth Manifolds*. New York, NY, USA: Springer, 2013.
[39] Z. Zhang, J. Su, E. Klassen, H. Le, and A. Srivastava, "Video-based action recognition using rate-invariant analysis of covariance trajectories," 2015, *arXiv:1503.06699*. [Online]. Available: http://arxiv.org/abs/1503.06699
[40] J. Su, A. Srivastava, F. D. M. de Souza, and S. Sarkar, "Rate-invariant analysis of trajectories on Riemannian manifolds with application in visual speech recognition," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 620–627.
[41] Z. Lin, R. Liu, and H. Li, "Linearized alternating direction method with parallel splitting and adaptive penalty for separable convex programs in machine learning," *Mach. Learn.*, vol. 99, no. 2, pp. 287–325, May 2015.
[42] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 612–620.
[43] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, Jan. 2010.
[44] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 123–231, 2013.
[45] M. Yin, J. Gao, Z. Lin, Q. Shi, and Y. Guo, "Graph dual regularized low-rank matrix approximation for data representation," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4918–4933, Dec. 2015.
[46] B. H. Williams, M. Toussaint, and A. J. Storkey, *Extracting Motion Primitives from Natural Handwriting Data*. Athens, Greece: Springer, 2006.
[47] B. Williams, M. Toussaint, and A. J. Storkey, "Modelling motion primitives and their timing in biologically executed movements," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1609–1616.
[48] T. E. de Campos, B. R. Babu, and M. Varma, "Character recognition in natural images," in *Proc. Int. Conf. Comput. Vis. Theory Appl.*, 2009, pp. 273–280.
[49] M. W. Kadous, "Temporal classification: Extending the classification paradigm to multivariate time series," Ph.D. dissertation, School Comput. Sci. Eng., Univ. New South Wales, Sydney, NSW, Australia, 2002.
[50] S. Tierney, Y. Guo, and J. Gao, "Selective multi-source total variation image restoration," in *Proc. Int. Conf. Digit. Image Comput., Techn. Appl. (DICTA)*, Nov. 2015, pp. 1–8.

Yi Guo received the B.Eng. degree (Hons.) in electrical engineering from the North China University of Technology, Beijing, China, in 1998, the M.Eng. degree in automatic control from Central South University, Changsha, China, in 2002, and the Ph.D. degree from the University of New England, Armidale, NSW, Australia, in 2008.

Since 2005, he has been studying computer science at the University of New England, focusing on dimensionality reduction for structured data with nonvectorial representation. From 2008 to 2016, he was with CSIRO, Canberra, ACT, Australia, where he was a Computational Statistician and worked on various projects in spectroscopy, remote sensing, and materials science. He joined Western Sydney University, Parramatta, NSW, Australia, in 2016. His recent research interests include machine learning, computational statistics, and optimization.

Stephen Tierney received the bachelor's degree (Hons.) in computer science and the Ph.D. degree from Charles Sturt University, Bathurst, NSW, Australia, in 2012 and 2017, respectively.

He is currently a Lecturer in business analytics with The University of Sydney, Sydney, NSW, Australia. His recent research interests include machine learning, data visualization, recommendation systems, and Bayesian optimization.

Junbin Gao received the B.Sc. degree in computational mathematics from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 1982, and the Ph.D. degree from the Dalian University of Technology, Dalian, China, in 1991.

He was a Professor of computer science with the School of Computing and Mathematics, Charles Sturt University, Bathurst, NSW, Australia. From 1982 to 2001, he was an Associate Lecturer, a Lecturer, an Associate Professor, and a Professor with the Department of Mathematics, HUST. He was a Senior Lecturer and a Lecturer in computer science with the University of New England, Armidale, NSW, Australia, from 2001 to 2005. He is currently a Professor of big data analytics with The University of Sydney Business School, The University of Sydney, Sydney, NSW, Australia. His current research interests include machine learning, data analytics, the Bayesian learning and inference, and time series.