Functional Subspace Clustering with Application to Time Series

Mohammad Taha Bahadori[*] MOHAMMAB@USC.EDU
David Kale[*†] DKALE@USC.EDU
Yingying Fan[*] FANYINGY@MARSHALL.USC.EDU
Yan Liu[*] YANLIU.CS@USC.EDU

[*]University of Southern California, Los Angeles, CA 90089
[†]Laura P. and Leland K. Whittier Virtual PICU, Children's Hospital Los Angeles, Los Angeles, CA 90027

Abstract

Functional data, where samples are random functions, are increasingly common and important in a variety of applications, such as health care and traffic analysis. They are naturally high dimensional and lie along complex manifolds. These properties warrant use of the subspace assumption, but most state-of-the-art subspace learning algorithms are limited to linear or other simple settings. To address these challenges, we propose a new framework called *Functional Subspace Clustering* (FSC). FSC assumes that functional samples lie in deformed linear subspaces and formulates the subspace learning problem as a sparse regression over operators. The resulting problem can be efficiently solved via greedy variable selection, given access to a fast deformation oracle. We provide theoretical guarantees for FSC and show how it can be applied to time series with warped alignments. Experimental results on both synthetic data and real clinical time series show that FSC outperforms both standard time series clustering and state-of-the-art subspace clustering.

1. Introduction

Classical machine learning models assume that each observation is represented as a finite dimensional vector. However, many applications involve *functional data*, where samples are random functions (instead of standard vectors) representing continuous processes and exhibiting structure (Ferraty & Romain, 2011). Functional data are increasingly common and important in a variety of scientific and commercial domains, such as healthcare, biology, traffic analysis, climatology, and video. As a result, many statistical methods for analyzing functional data have been proposed (Müller, 2011; Hall & Hosseini-Nasab, 2009; Hall, 2011).

Functional data present challenges and opportunities for machine learning, especially in clustering and representation learning. The underlying process is a continuous function of infinite dimension, usually unknown and difficult to represent directly. Even when only finite samples are available, they can be difficult to work with. Time series, for example, exhibit noise, different lengths, and irregular sampling. Thus, the first step in functional data clustering is often to transform the data into a more regular representation (Hall, 2011; Delaigle et al., 2012; Shang, 2013) to which standard clustering can be applied (e.g., k-means). Alternative non-parametric approaches define a measure of similarity between samples and cluster in the similarity (or *affinity*) space (Warren Liao, 2005; Cuturi, 2011). Such approaches often utilize specialized measures of similarity that provide invariance to transformations or *deformations*. In object recognition, images of the same object should be similar regardless of resolution, lighting, or angle. In time series data mining, *dynamic time warping* (DTW) is used to compare time series based on shape and permits distortions (e.g., shifting and stretching) along the temporal axis, as shown in Fig. 1 (Vintsyuk, 1968). Such similarities can often be used to perform effective clustering (Warren Liao, 2005; Petitjean et al., 2014) but are not immune to the curse of dimensionality inherent in functional data (Ferraty & Vieu, 2006; Geenens, 2011). What is more, they can produce complex manifolds difficult to model using classic dimensionality reduction techniques (e.g., PCA) and cluster models (Vidal, 2011).

Subspace clustering, an increasingly popular technique in machine learning, addresses many of the aforementioned challenges. Subspace clustering can capture more complex manifolds and is robust in higher dimensional settings (Vidal, 2011; Kriegel et al., 2012), both desirable properties in practical applications. In health care, for example, hos-

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

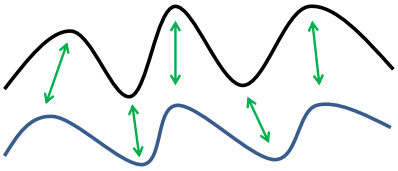


Figure 1. Illustration of the deformation operation for functional data. Two functions are considered similar if a deformation of one of them is similar to the other one. The figure has been regenerated from (Müller, 2007, Fig. 4.1).

pitalized patients with different underlying diseases (i.e., clusters) often exhibit shared or overlapping sets of symptoms (i.e., subspaces). However, most state-of-the-art subspace clustering algorithms with provable guarantees are limited to linear settings, making them infeasible for functional data and incompatible with deformation-based similarity measures.

In this paper, we propose a new clustering framework for functional data called *Functional Subspace Clustering* (FSC). FSC extends the power and flexibility of subspace clustering to functional data by permitting the deformations that underlie many popular functional similarity measures. The result is a framework that differs from most existing approaches to functional data clustering. In particular, FSC does not assume a structured generative model (e.g., a sequential model for time series) or a predefined set of basis functions (e.g., B-splines). The FSC framework, described in Section 3, works as follows: first, we define a subspace model which allows the functional data to come from multiple deformed linear subspaces. Then we formulate the subspace learning problem as a *sparse subspace clustering problem*, similar to (Elhamifar & Vidal, 2013) but as an optimization over operators. Finally, we introduce an efficient learning algorithm, based on greedy variable selection and assuming access to a fast oracle that can return the optimal deformation between two functions.

We provide theoretical guarantees for FSC with a general class of deformations (Section 3.3). In Section 4, we apply FSC to a common functional data setting: time series with warped alignments. We provide an efficient implementation of the warping oracle and show how this algorithm can also be used to retrieve the learned basis functions for each deformed subspace. These bases can be used as lower-dimensional features for either classification or clustering. Experimental results on synthetic data and two real hospital data sets, described in Section 5, show that FSC significantly outperforms both standard time series clustering and state-of-the-art subspace clustering. In clinical data, our framework learns physiological patterns that can be used to discriminate patients based on risk of mortality.

2. Related Work

Functional clustering There has been a significant amount of research on functional data clustering. This is commonly performed using a two step process, in which functions are first mapped into a fixed size representations and then clustered. For example, we can fit the data to predefined base functions, such as splines or wavelets (Wang et al., 2007). In time series data mining, researchers often use *motifs* or common patterns discovered from the data (Lin et al., 2007). There is a growing body of literature on models for directly clustering functional data without the two-step process (James & Sugar, 2003; Jacques & Preda, 2014). These approaches sometimes make strong assumptions about the underlying function or ignore important structure, such as time order (Hall, 2011).

Nonparametric clustering methods are popular in the data mining literature, where researchers combine specialized distance metrics with simple clustering methods. Functional distance metrics allowing deformation date back several decades (Vintsyuk, 1968; Sakoe & Chiba, 1978). However, it has been shown only recently in the functional data analysis literature that deformation-based metrics can be more robust to the curse of dimensionality than simple Euclidean distance (Ferraty & Vieu, 2006; Geenens, 2011). For time series, DTW is a popular technique for measuring the distance between two time series with temporal deformations (Vintsyuk, 1968; Sakoe & Chiba, 1978; Müller, 2007). Given the distance metric, we can use k-means directly (Petitjean et al., 2014) or construct an affinity matrix and apply spectral clustering (Rakthanmanon et al., 2012).

Subspace Clustering Unlike much existing work on time series clustering, FSC is based on subspace clustering. Subspace clustering is a generalization of PCA that can discover lower dimensional representations for multiple principal subspaces, enabling it to model more complex manifolds (Vidal, 2011). Subspace clustering is a common tool for cluster analysis in high dimensional settings (Kriegel et al., 2012). Both of these properties make it well-suited for functional data. Sparse subspace clustering (SSC) does not require local smoothness, permitting disparate points to constitute subspaces (Elhamifar & Vidal, 2009). It formulates subspace learning and neighbor selection as a regression, and admits a variety of efficient solutions based on LASSO (Elhamifar & Vidal, 2009; 2013; Soltanolkotabi et al., 2014), thresholding (Heckel & Bölcskei, 2013; Heckel et al., 2014), and greedy orthogonal matching pursuit (Dyer et al., 2013; Park et al., 2014). SSC has strong theoretical guarantees and is robust to outliers, which are common in functional data.

Alternative Approaches FSC does not assume any particular sequential generative process, as in (Afsari & Vidal, 2014; Jebara et al., 2007; Kim & Smyth, 2006), or a

predefined set of basis functions, such as B-splines, Bezier curves, or truncated Fourier functions (Gaffney & Smyth, 2004; Saria et al., 2011; Yuan & Li, 2014). FSC also admits a theoretical analysis, unlike many of alternative frameworks. (Yuan & Li, 2014) propose a subspace clustering framework for images that uses predefined truncated Fourier basis functions to implicitly capture different kinds of image deformations. They enumerate all possible deformed bases and then apply Group LASSO to learn the affinity matrix. However, this strategy does not generalize to many functional data problems where the space of potential deformations can be too large to enumerate explicitly, such as warped alignments between time series. In contrast, FSC does not require explicit enumeration or representation of the deformations. Instead, it makes use of the fast deformation oracles that have been proposed and studied for many common function data problems (e.g., DTW for warping distance in time series). Combined with simple greedy variable selection, this makes FSC computationally more efficient than the Group LASSO formulation in (Yuan & Li, 2014).

3. Functional Subspace Clustering

In this section, we present our proposed *Functional Subspace Clustering* (FSC) algorithm and elucidate the challenges that functional data present to traditional subspace clustering methods. We first discuss our data model and assumptions in Section 3.1, and then we describe the FSC framework in Section 3.2 and provide a theoretical analysis in Section 3.3. We will discuss how FSC can be applied to time series data with warping in Section 4.

3.1. Data Model

Let $X_1, \ldots, X_n$ denote n functions on a compact interval $\mathscr{I}$, such that $\int_{\mathscr{I}} \mathbb{E}[X_i^2] < \infty$ for $i = 1, \ldots, n$. We observe noisy instances as follows

$$Y_i = X_i + \varepsilon_i, \quad \text{for } i = 1, \ldots, n. \tag{1}$$

where ε_i for $i = 1, \ldots, n$ are i.i.d. instances from a random function with zero mean and $\int_{\mathscr{I}} \mathbb{E}[\varepsilon_i^2] = \sigma^2$.

Subspace Assumption The functions (curves) X_i are selected from L manifolds S_ℓ for $\ell = 1, \ldots, L$. Given a set of basis functions Φ_ℓ, each manifolds S_ℓ is defined as the set of all functions that are deformation (warping) of linear combinations of basis functions in Φ_ℓ:

$$S_\ell \triangleq \left\{ X \;\middle|\; X = d\Big(\sum_{\phi_k \in \Phi_\ell} \alpha_k \phi_k \Big);\ \alpha_k \in \mathbb{R},\ d \in \mathcal{D} \right\}, \tag{2}$$

where ϕ_k are the basis functions and the set $\mathcal{D}$ contains all possible deformation operators d. We denote the set of all given functions that belong to a manifold S_ℓ with $\mathcal{X}_\ell$ and the corresponding noisy observation sets by $\mathcal{Y}_\ell$. Our main goal is to group $X_1, \ldots, X_n$ according to their corresponding subspaces as defined in Eq. (2).

While the sets defined in Eq. (2) are not linear subspaces in general, they show similar properties under appropriate conditions. In particular, suppose the set of deformations are linear maps and form a finite group with group law defined as the composition operation. The group assumption requires that composition of two deformations belong to the set $d_1 \circ d_2 \in \mathcal{D}$ for every $d_1, d_2 \in \mathcal{D}$ and for every $d \in \mathcal{D}$ there exists an inverse operation d^{-1} such that $d^{-1}(d(X)) = X$ for every $X \in S_\ell$. The permutation groups are prominent groups satisfying these conditions. We can show that under this assumption, every function in the manifold with s basis functions can be written as linear combination of s or more other functions in the same manifold. Specifically, for every $X_i \in S_\ell$, we can write the following generalization of the self-expressive equation

$$X_i = \sum_{X_j \in S_\ell, j \neq i} \beta_j d_j(X_j), \tag{3}$$

with some deformation $d_j \in \mathcal{D}$ and scalars $\beta_j \in \mathbb{R}$. A proof is provided in Appendix A. Note that our algorithm will not rely on these assumptions to operate; for example it will not need to compute the inverse of a deformation. FSC can be applied to any data for which the self-expressive property in Eq. (3) holds.

3.2. Functional Subspace Clustering

Given the result in Eq. (3), the cluster assignments of the functional data generated according to Eq. (2) can be uncovered using a novel variant of sparse subspace clustering. We solve the following sparse regression problem for all functions $Y_1, \ldots, Y_n$:

$$\begin{aligned} \widehat{B}_{i,:} = \underset{B_{i,:}, \{d_j\}}{\operatorname{argmin}} & \left\| Y_i - \sum_{j \neq i} B_{i,j} d_j(Y_j) \right\|_2^2, \\ \text{subject to} \quad & \|B_{i,:}\|_0 \leq s. \end{aligned} \tag{4}$$

where $B \in \mathbb{R}^{n,n}$. The L_0 sparsity pseudo-norm indicates the number of non-zero elements of a vector. The goal of this regression is to find the best sparse approximation for Y_i by selecting a few functions Y_j, deforming them by optimizing d_j, and scaling them by multiplying with $B_{i,j}$. After solving Eq. (4) for all functions, similar to subspace clustering we define the symmetric affinity matrix $\boldsymbol{A} = |\boldsymbol{B}| + |\boldsymbol{B}|^\top$ and apply spectral clustering (Ng et al., 2002; Von Luxburg, 2007) on $\boldsymbol{A}$, described in Algorithm 2 in Appendix C. We can also compute the Laplacian embedding to extract a lower dimensional representation of the functions, useful for other machine learning tasks (e.g., classification) (Schölkopf & Smola, 2002, Chapter 14).

Algorithm 1: Functional subspace clustering.

Data: Noisy functional observations $\{Y_i\}_{i=1}^n$ and a termination criteria ϵ.
Result: Clustering assignments for Y_i, $i = 1, \dots, n$.

for $i = 1, \dots, n$ **do**
  Initialize $\mathcal{F} \leftarrow \varnothing$, $\mathcal{J} \leftarrow \{i\}$, $R_1 \leftarrow Y_i$, $l \leftarrow 1$
  while $\max_{j \notin \mathcal{J}, d_j} \frac{|\langle R_l, d_j(Y_j)\rangle|}{\|d_j(Y_j)\|_2 \|R_l\|_2} > \epsilon$ **do**
    $\widehat{\boldsymbol{\phi}}_j \leftarrow \operatorname{argmax}_{j \notin \mathcal{J}, d_j} \frac{|\langle R_k, d_j(Y_j)\rangle|}{\|d_j(Y_j)\|_2}$
    $\mathcal{F}_l \leftarrow \mathcal{F}_{l-1} \cup \{\widehat{\boldsymbol{\phi}}_j\}$
    $\mathcal{J} \leftarrow \mathcal{J} \cup \{j\}$
    $\widehat{B}_{i,:} \leftarrow \operatorname{argmin}_{B_{i,:}} \|Y_i - \sum_{\boldsymbol{\phi}_j \in \mathcal{F}_l} B_{i,j} \boldsymbol{\phi}_j\|_2^2$
    $R_{l+1} \leftarrow Y_i - \sum_{\boldsymbol{\phi}_j \in \mathcal{F}_l} \widehat{B}_{i,:} \boldsymbol{\phi}_j$
    $l \leftarrow l + 1$
  end
end
$\boldsymbol{A} \leftarrow |\boldsymbol{B}| + |\boldsymbol{B}|^\top$
Apply spectral clustering to $\boldsymbol{A}$ (e.g., Algorithm 2 in Appendix C) to obtain cluster assignments.

Unlike the linear sparse subspace clustering setting, Eq. (4) is a large-scale sparse regression which requires optimization over both $\boldsymbol{B}$ and d. The optimization over the deformation operator can be especially difficult, as it is an operator optimization.

Fast Sparse Regression with an Oracle Our approach for efficiently solving Eq. (4) is summarized in Algorithm 1 and is based on three main steps: a relaxation to a regular sparse linear regression problem, use of a fast oracle to find the best deformation, and then greedy variable selection. In the first step, we consider all possible deformations of each Y_j as covariates in the regression. This relaxation makes the problem linear and convex but introduces a new computational challenge: it dramatically increases the dimensionality of the regression. For example, given two time series $Y_1 \in \mathbb{R}^{T_1}$ and $Y_2 \in \mathbb{R}^{T_2}$, there are $\mathcal{O}(\exp(T_1 + T_2))$ possible warping-based alignments. Merely enumerating all possible warpings and updating the gradient becomes computationally expensive, and solving Eq. (4) becomes practically intractable.

We address this computational bottleneck by assuming that we have access to a fast oracle that can identify the best deformation for any pair of functions Y_1 and Y_2, defined as $d^\star = \operatorname{argmax}_d |\langle Y_1, d(Y_2)\rangle| / \|d(Y_2)\|_2$. Now rather than solving a complex nonlinear regression or enumerating all possible deformations, we can simply query the oracle for the best deformation for each Y_j. Computationally, availability of this oracle significantly simplifies the sparse regression problem and yields an efficient algorithm for solving Eq. (4). DTW is an example of such an oracle for measuring time series similarity, especially when it is combined with constraints and early-stopping heuristics (Rakthanmanon et al., 2012).

With such a fast oracle available, we can use greedy variable selection with orthogonal projections (OMP) to solve Eq. (4) efficiently, as it only requires a limited number of calls to the oracle to solve the sparse regression problem. We could also use the thresholding approach for sparse subspace clustering, as in (Heckel & Bölcskei, 2013; Heckel et al., 2014), but as the authors note and we also confirm in the experiments, the greedy approach typically has better empirical performance. The full algorithm for solving Eq. (4) is shown in Algorithm 1.

3.3. Analysis

One major advantage of sparse subspace clustering is that we can find conditions under which the success of the algorithm is guaranteed. We begin by defining two quantities to describe the similarity of subspaces and the density of points in each cluster, respectively. To measure subspace similarity, we define the principal angle between two deformed subspaces S_ℓ and $S_{\ell'}$ as

$$\theta_{\ell,\ell'} = \arccos \sup_{V \in S_\ell, U \in S_{\ell'}, d, d'} \frac{|\langle d(V), d'(U)\rangle|}{\|d(V)\|_2 \|d'(U)\|_2},$$

where the supremum is over functions V and Z from S_ℓ and $S_{\ell'}$, respectively, and over their respective deformations. We also define the minimum principal angle $\theta_\ell = \min_{\ell' \neq \ell} \theta_{\ell,\ell'}$ over all pairs of subspaces, in order to provide a uniform bound for all ℓ. To measure cluster density, we define the covering radius r_ℓ as

$$r_\ell = \max_{Y_j \in \mathcal{Y}_\ell} \max_{V \in S_\ell} \min_{Y \in \{\mathcal{Y}_\ell \setminus Y_j\}} \operatorname{dist}(V, Y),$$

where $\operatorname{dist}(V, Y) = \sup_{d,d'} \sqrt{1 - \frac{|\langle d(V), d'(Y)\rangle|^2}{\|d(V)\|_2 \|d'(Y)\|_2}}$. The following theorem provides a sufficient condition for success of our algorithm in the noiseless setting:

Theorem. *For any function $Y_i \in \mathcal{Y}_\ell$, Algorithm 1 will select only the functions from the same subspace as Y_i's neighbors if the termination criterion is $\epsilon > \cos(\theta_\ell)(1 + \sqrt{2} r_\ell)$.*

The full proof is provided in Appendix B. The theorem's main implication is that the algorithm will not make any mistake in identifying Y_i's cluster neighbors, provided that the subspaces are sufficiently different and each cluster is sufficiently large. Thus, if the algorithm finds enough neighbors for each data point, then the functions in each subspace will create connected clusters, ensuring the success of spectral clustering.

Discussion One implication of this theorem is that we need to control the flexibility of the deformation operator

because of its influence on the performance of FSC. Excessive flexibility increases the overlap of (i.e., decreases the principal angle between) the subspaces, which can degrade performance. Furthermore, we need to restrict the number of possible deformations to be polynomial in order to ensure asymptotic consistency of our variable selection algorithm. Thus, we need to carefully manage the flexibility of the deformation operator. In time series analysis, for example, it is common to use a constrained warping window with DTW (Sakoe & Chiba, 1978).

4. Functional Subspace Clustering of Time Series with Warping (FSC-TW)

In this section, we show how FSC can be applied to time series data with warping-based alignment deformations. An alignment deformation (or *warping function*) d maps the samples of one time series Y_i onto those of a second time series Y_j, while preserving the time order. We assume that we are given only a finite set of observations from each Y_i, indexed as Y_{it} for $t \in \mathcal{T}_i \subset \mathscr{I}$ where $0 < |\mathcal{T}_i| < \infty$. Thus, an alignment is typically realized as a list of non-decreasing pairs of indices with constraints on neighboring pairs (e.g., each index can change by at most one from one pair to the next). Given a measure of discrepancy (or similarity) between individual time points, the minimum warping distance (or maximum warping similarity) can be computed in quadratic time using dynamic programming. This is known as *dynamic time warping* (DTW) (Vintsyuk, 1968). The set of all warping deformations is not a group as it does not satisfy the conditions in Section 3.2, nevertheless we can still perform approximate clustering via FSC with time warping deformations.

DTW is, in principle, a fast oracle for returning the best warping alignment between two time series, but because it computes an un-normalized distance (making distances between different pairs of time series incomparable), it cannot be used with FSC in practice. Thus, we develop an alternative algorithm for quickly computing the optimal alignment between two time series, which returns a normalized distance and can be used as a deformation oracle for FSC. We describe this algorithm in Section 4.1 and then show in Section 4.2 how its formulation can be used to recover the latent basis functions learned for time series under warping deformations.

4.1. Fast Warping Selection for Time Series

Here we develop an alternative oracle for efficiently selecting the best warping between two time series. We begin by observing that during greedy variable selection in Algorithm 1, the best direction is given as $\tilde{Y}_j = \operatorname{argmax}_{d(Y_j)} \frac{(\langle R_i, d(Y_j)\rangle)^2}{\|d(Y_j)\|_2^2}$ where we have used R_i to denote the residual of Y_i at some iteration of Algorithm 2. For simplicity, we assume that the length of the time series R_i and Y_j are equal to T_1 and T_2, respectively. In order to efficiently find the optimal warping for a time series Y_i, we note that every warping is an assignment of each point in Y_j to one point in R_i. Thus, we can use a list of binary indicator vectors $Z = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{T_1})$, $\boldsymbol{z}_k \in \{0,1\}^{T_2}$ to represent every deformation as $d(Y_j) = (\boldsymbol{z}_1^\top Y_j, \ldots, \boldsymbol{z}_{T_1}^\top Y_j)$. Now we can reformulate the warping selection process as an integer program

$$\{\boldsymbol{z}_k^\star\} = \underset{\{\boldsymbol{z}_k\}}{\operatorname{argmin}} \frac{\sum_{k=1}^{T_1} (\boldsymbol{z}_k^\top Y_j)^2}{\left(\sum_{k=1}^{T_1} R_{ik} \boldsymbol{z}_k^\top Y_j\right)^2} \qquad (5)$$

$$\text{s.t.} \quad z_{k,\ell} \in \{0,1\}, \quad \sum_{\ell}^{T_2} z_{k,\ell} = 1.$$

We impose additional linear constraints to guarantee that the warping preserves the time order. In particular, if Y_{jt} is assigned to $R_{it'}$, Y_{js} is assigned to $R_{is'}$, and $t < s$, we require $t' \leq s'$. To enforce this constraint, it suffices to consider the integer number $\bar{\boldsymbol{z}}$ corresponding to the binary vector $\boldsymbol{z}$ and require $\bar{\boldsymbol{z}}_k \leq \bar{\boldsymbol{z}}_{k+1}$ for $k = 1, \ldots, T_1 - 1$. This constraint can be implemented as a set of linear constraints by considering the positional binary notation. In practice, we also restrict the warping to not map data points that are apart from each other by more than Δ time stamps (Sakoe & Chiba, 1978). Implementing this constraint reduces the number of variables in the optimization from $T_1 T_2$ to $T_1(2\Delta + 1)$ which can accelerate the algorithm if the time series are long. Notice that the optimization in Eq. (5) can be readily used in irregular and multivariate time series, as well.

Relaxing the integer constraint in Eq. (5), the problem becomes convex and can be efficiently solved. Given the fact that we need to solve Eq. (5) for all time series $Y_j, j \neq i$, we use Frank-Wolfe's algorithm (Jaggi, 2013) because it provides an inexpensive certificate for the duality gap in each iteration of the optimization problem. We can use this to disqualify suboptimal directions during the greedy search in Algorithm 1 by checking the current search direction's duality gap against the previous optimum. Given the simple form of Eq. (5) and the fact that it is strongly convex, we further accelerate the optimization by performing a line search on a grid of step size values, followed by a few iterations of Newton's method. In practice, the optimization converges within very few iterations.

4.2. Identifying the Latent Basis Functions

The formulation of warping operator in Eq. (5) enables us to recover the deformed latent functions by solving Eq. 6. Without loss of generality, suppose $Y_1, \ldots, Y_m$ have been clustered into a single cluster and that the underlying latent

space is r dimensional. We need to solve the following optimization problem (PCA-TW)

$$\min_{\boldsymbol{Z},\boldsymbol{\phi},\boldsymbol{W}} \sum_{i=1}^{m} \left\| Y_i - \sum_{k=1}^{r} Z_{i,k}\boldsymbol{\phi}_k w_{i,k} \right\|_2^2 \quad (6)$$
$$\text{s.t. } Z_{i,k} \in \mathcal{L},\ \|\boldsymbol{\phi}_k\|_2 = 1.$$

where $\mathcal{L}$ denotes the set of constraints described in Section 4.1. We need to solve for the bases $\boldsymbol{\phi}_k$, alignment variables $Z_{i,k}$, and weights $w_{i,k}$. To solve this problem, we alternate between optimizing over $\boldsymbol{\phi}_k$ and $\{w_{i,j}, Z_{i,k}\}$, while fixing the other one. For learning the $\boldsymbol{\phi}_k$, we initialize them using PCA and optimize the loss function together with the fused lasso regularizer (Tibshirani et al., 2005) to obtain a smooth function. For optimization over $Z_{i,k}$ and $w_{i,k}$, analytical solution of optimization over $w_{i,k}$ leads to an optimization problem similar to Eq. (5) for $Z_{i,k}$ which can be solved by the same method described in the previous section.

Discussion The formulation of deformation in Eq. (5) is not limited to time warping but encompasses many existing deformation operations discussed in the literature. For example, two time series may be considered similar if only certain subsequences are similar. We can capture this phenomenon by relaxing the constraint $\sum_{\ell}^{T_2} z_{k,\ell} = 1$ to $\sum_{\ell}^{T_2} z_{k,\ell} \leq 1$. The formulation in Eq. (5) is also appropriate for handling the missing data settings. Note that because of the subspace clustering nature of the algorithm, smoothness of the time series will be automatically incorporated in the clustering process.

5. Experiments

To demonstrate the effectiveness of FSC, we perform experiments using one synthetic dataset and two real world datasets related to health and wellness. All data are time series, so we use the *time series with warping* variant of our algorithm, FSC-TW. We compare FSC-TW's performance to the following baselines:

ED+SC We apply spectral clustering to an affinity matrix based on Euclidean distance, created as follows: first, we construct a distance matrix $\boldsymbol{D}$ and normalize it by its largest element. Then we define $\boldsymbol{A} = \exp(-\boldsymbol{D}) + \exp(-\boldsymbol{D}^\top)$ and apply spectral clustering to $\boldsymbol{A}$.
DTW+SC We apply spectral clustering to a DTW-based affinity matrix, constructed using the same procedure.
GAK+SC We apply spectral clustering to an affinity matrix constructed using the Global Alignment Kernel (GAK) (Cuturi et al., 2007; Cuturi, 2011), a variant of DTW that yields a valid positive semidefinite kernel.
SSC We apply the original Sparse Subspace Clustering algorithm proposed in (Elhamifar & Vidal, 2009), without deformations.
TSC-TW We apply the Thresholded Subspace Clustering (TSC) algorithm from (Heckel et al., 2014), combined with our warping deformation oracle from Eq. (5).

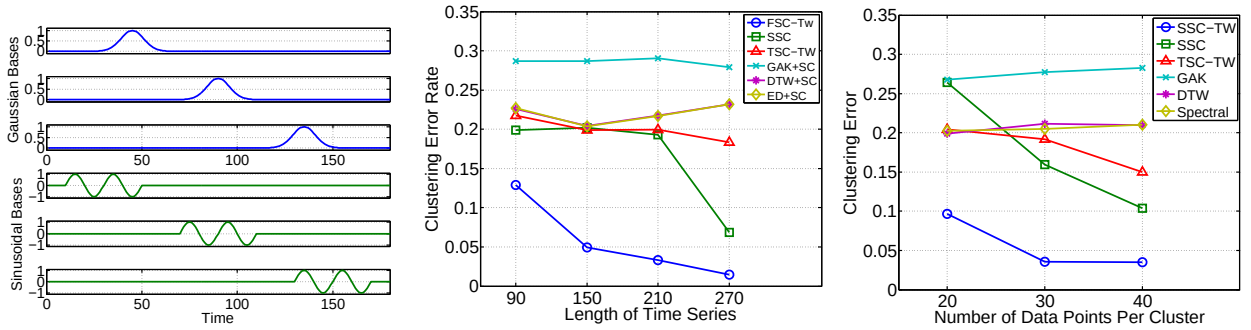
The ED+SC, DTW+SC, and GAK+SC baselines enable us to evaluate the benefit provided by sparse subspace clustering, in comparison to simple deformation-based clustering. The SSC baseline allows us to determine whether allowing deformed subspaces improves the performance of subspace clustering. The TSC-TW provides a comparison with an alternative sparse subspace clustering with time warping and another variable selection technique.

5.1. Synthetic Data Experiments

We begin with synthetic data experiments to investigate how FSC-TW performs on data generated from the assumed deformed subspace model described in Section 3.1 and to explore the impact of data dimensionality, subspace separation, and cluster density. First, we generate two synthetic datasets with different basis time series, shown in Fig. 2(a). We create three subspaces, each including two of the basis functions. We use two forms of deformation operators: (i) a random shift in time, selected uniformly from $[-10, 10]$ and (ii) time warping with maximum window of length 10. We then investigate how increasing both the length of the synthetic time series and the number of points per cluster impacts the cluster error rate of the different algorithms. The results in Fig. 2 confirm the utility of the deformed subspace assumption and the superior performance of FSC-TW.

Comparing the baseline algorithms, we can divide them into two categories: subspace clustering based algorithms (FSC-TW, TSC, SSC) and regular spectral clustering with different distance metrics. Given the true subspace model in the synthetic data, we expect the first group to perform better. Among the algorithms in the first group, SSC does not capture the deformations in the data. While TSC-TW captures deformations, subspace clustering with thresholding is empirically shown to have inferior performance compared to sparse subspace clustering (Heckel & Bölcskei, 2013). Thus, we expect FSC-TW to perform superior compared to SSC and TSC-TW.

In Fig. 2(b) we fix the number of examples in each cluster to 50 and evaluate performance as the length of the time series increases. Increasing the length of the time series increases the dimensionality of the data, which in turn increases the sparsity of each point's neighborhood and the separation of the subspaces; i.e. the principal angles θ_ℓ in Section 3.3 increases. As expected, the error for all subspace clustering algorithms improve as length increases, while the performance of non-subspace methods gradually degrades. This is consistent with two previous findings: first, that DTW provides minimal advantage over ED for



(a) Bases (b) Length of Time Series (c) Number of Time Series

Figure 2. Synthetic data experiments. (a) The bases used for constructing the synthetic data. (b,c) The clustering error rate for six algorithms as (b) the length of time series grows and (c) the number of time series per cluster grows.

long time series (Lin et al., 2012); and second, that subspace clustering is especially beneficial in higher dimensions (Kriegel et al., 2012). FSC-TW outperforms the baselines at all tested lengths.

In Fig. 2(c) we fix the length of the time series to 150 and increase the number of data points in each cluster, which also increases the density of points within each cluster and potentially increasing overlap between clusters and a more complicated neighborhood structure. The overall trend is similar to that of length: FSC-TW is clearly superior for all sizes, and the subspace cluster methods improve rapidly as the clusters grow in size. Again, this is consistent with what is known about DTW (it provides less benefit in large scale time series datasets (Lin et al., 2012)) and about subspace clustering. As we increase the number of data points per cluster, the probability that subspace clustering finds the correct clustering increases because the probability that two points from the same cluster are subspace neighbors increases. Once again, FSC-TW outperforms the baselines.

It is very interesting that plain SSC becomes increasingly robust to deformations as the time series become longer *and* as the data set size grows. This suggests that the subspace model assumptions are well-suited to functional data, at least under these conditions. However, it performs quite poorly for small numbers of short time series. FSC-TW is robust to length, yielding the best performance for both short and long time series. Together, these results suggest that our combination of subspaces, deformations, and greedy variable selection yields a powerful clustering framework for functional data.

5.2. Real World Data

We apply FSC-TW and our baselines to two real world data sets related to health:

ICU This is a collection of multivariate clinical time series extracted from a major hospital's electronic health records (EHRs) system, recorded by clinical staff during care in an intensive care unit (ICU). Each time series includes 24 hours of measurements for 13 variables, including vital signs, lab tests, or clinical observations, with one observation per hour. In these data, subspaces correspond to collections of physiologic signs and symptoms, while clusters represent cohorts of similar patients. We treat in-hospital mortality prediction as a binary classification task.

Physionet The Physionet dataset[1] is a publicly available collection of multivariate clinical time series, similar to ICU but with additional variables. The time series are also 48 hours long and include in-hospital mortality as a binary label. Fig. 4 in Appendix C shows that the two classes are very difficult to distinguish on the basis of their raw time series data alone.

In all of the datasets, we normalize each time series to have zero mean and unit variance. Then, we apply each algorithm to learn the affinity matrix and then extract lower dimensional representations as described in Algorithm 2 in Appendix C. We then evaluate the utility of these representations by using them as features in a RBF-SVM binary classifier. For evaluation, we create 30 randomly divided training and testing partitions. For each partition, we train the RBF-SVM on the training partition set using 5-fold cross validation, then test it on the corresponding test set. Table 1 summarizes the AUC for each method and data set, averaged across partitions.

Results The results in Table 1 reveal several interesting trends. First, FSC-TW once again yields the best classifier, and only FSC-TW and SSC yield a classifier that is statistically different from guessing at random in both datasets, validating the sparse subspace assumption. This confirms our intuition that the complex latent manifold structure of critical illness (in terms of symptoms and signs) is captured better by subspace clustering than by simpler methods. What is more, the subspace assumption alone may provide some robustness to deformations, as observed in

[1] http://physionet.org/challenge/2012/

Table 1. Average AUC obtained by the algorithms on the real world datasets.

Dataset	FSC-TW	TSC-TW	SSC	GAK+SC	DTW+SC	ED+SC
ICU	$\mathbf{62.32 \pm 8.37}$	56.54 ± 9.78	61.99 ± 7.89	56.35 ± 8.06	59.55 ± 8.17	58.76 ± 9.83
Physionet	$\mathbf{66.27 \pm 6.08}$	52.41 ± 6.61	62.51 ± 8.56	51.30 ± 7.99	50.56 ± 7.99	49.73 ± 7.25

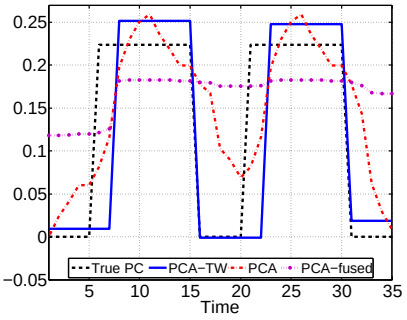


Figure 3. Synthetic data experiment: PCA-TW is the only algorithm that successfully recovers the principal component under deformation.

the synthetic data results.

The more interesting trend is the interaction between subspaces and warping. Using a warping-based distance benefits subspace clustering more than spectral clustering; we can see this when we compare the improvements in FSC-TW (vs. SSC) and DTW+SC (vs. ED+SC). This suggests that different patients may have conditions with different time courses in symptoms and treatment responses. However, we observe that the performance gains when adding warping for the clinical data sets are smaller than those observed in the synthetic data. Our hypothesis is that the degree of warping (shifts and stretches) in the real world clinical data may be relatively small (i.e., an hour or two) with respect to the hourly sampling rate.

Finally, we note that this is a very challenging classification problem: the patient outcome (i.e., death or discharge) can occur anywhere from hours to weeks after admission, but we are considering only the first 48 hours of data (Silva et al., 2012). Also, the outcome often depends upon a complex set of factors beyond initial presentation, including treatments, which are not available in these data and may occur after the first 48 hours (Paxton et al., 2013). What is more, the natural cluster structure is likely less correlated with outcome than it is with disease. Mortalities appear as outliers, rather than as a coherent cluster. In future work, we would like to apply FSC-TW to data with diagnostic labels to examine whether subspaces and clusters reflect known disease patterns.

5.3. Deformed basis function recovery

Next, we demonstrate PCA-TW's ability to learn and recover deformed basis functions (described in Section 4.2). We first demonstrate this using synthetic data, as follows: first, we select a principal vector (the dashed black line in Fig. 3). Then we generate 25 time series that are randomly shifted versions of this principal component. We then apply the algorithms to identify the true principal component, including basic PCA (red dashed line), our PCA-TW algorithm (solid blue line, Section 4.2), and PCA with a fused LASSO regularizer (PCA-fused, purple dashed line), which is also used by our algorithm. Figure 3 clearly shows that PCA-TW is the only algorithm that recovers the true shape of the basis and that its performance is not solely due to its use of the fused LASSO regularizer.

One of the main advantages of the proposed latent function learning algorithm in Section 4.2 is that it preserves more variance than the regular principal component analysis. At the same time, it is able to capture the main trend in the functional data without overfitting to the particular realization of the time series. The deformation allows us to obtain a principal component that preserves a larger amount of variance. The fraction of variance preserved in the first component by our algorithm is 30.52% and 39.36%, compared to 19.44% and 24.30% by PCA, for survivals and mortalities, respectively.

6. Conclusion and Future Work

We proposed *Functional Subspace Clustering* (FSC), a nonparametric functional clustering framework that can be applied to functional data with complex subspace structures and used with general deformation operations, including time series with warped alignments. We showed that this can be formulated as a sparse subspace clustering problem and solved using an efficient greedy algorithm with theoretical guarantees. Applied to time series data, FSC outperforms both standard time series clustering and linear subspace clustering.

While we provided a theoretical discussion about geometric properties of FSC, several theoretical questions remain unanswered. For example, under a random subspace model, what are the theoretical conditions for successful clustering? Also, while both the greedy variable selection and our deformation oracle algorithms are efficient, we can further accelerate their speed. We are interested in finding ways to scale up FSC. One possibility is to use an approximate fast oracle for selecting the greedy direction update.

Acknowledgment

The authors are grateful to Shay Deutsch and Michael Hankin for insightful discussions and to the anonymous reviewers for their constructive feedback. The authors would also like to thank Randall Wetzel of Children's Hospital Los Angeles for his assistance in analyzing clinical data. Mohammad Taha Bahadori was supported by NSF award number IIS 1254206. David Kale was supported by the Alfred E. Mann Innovation in Engineering Doctoral Fellowship, and the VPICU was supported by grants from the Laura P. and Leland K. Whitter Foundation. Yingying Fan was supported by NSF CAREER Award DMS-1150318. Yan Liu was supported by NSF IIS-1134990 and IIS-1254206 awards. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agency, or the U.S. Government.

References

Afsari, Bijan and Vidal, Ren. Distances on spaces of high-dimensional linear stochastic processes: A survey. In Nielsen, Frank (ed.), *Geometric Theory of Information*. Springer International Publishing, 2014.

Cuturi, Marco. Fast global alignment kernels. In *ICML*, pp. 929–936, 2011.

Cuturi, Marco, Vert, J-P, Birkenes, Øystein, and Matsui, Tomoko. A kernel for time series based on global alignments. In *ICASSP*, volume 2, pp. II–413. IEEE, 2007.

Delaigle, A., Hall, P., and Bathia, N. Componentwise classification and clustering of functional data. *Biometrika*, 99(2):299–313, April 2012.

Dyer, Eva L., Sankaranarayanan, Aswin C., and Baraniuk, Richard G. Greedy feature selection for subspace clustering. *JMLR*, 14:2487–2517, 2013.

Elhamifar, Ehsan and Vidal, René. Sparse subspace clustering. In *CVPR*, 2009.

Elhamifar, Ehsan and Vidal, René. Sparse subspace clustering: algorithm, theory, and applications. *PAMI*, 35 (11):2765–81, 2013.

Ferraty, Frederic and Romain, Yves. *The Oxford Handbook of Functional Data Analysis*. Oxford University Press, 2011.

Ferraty, Frédéric and Vieu, Philippe. *Nonparametric functional data analysis: theory and practice*. Springer, 2006.

Gaffney, Scott J and Smyth, Padhraic. Joint probabilistic curve clustering and alignment. In *NIPS*, 2004.

Geenens, Gery. Curse of dimensionality and related issues in nonparametric functional regression. *Statistics Surveys*, 5, 2011.

Hall, Peter. Principal component analysis for functional data: methodology, theory and discussion. In *The Oxford handbook of functional data analysis*, chapter 8. 2011.

Hall, Peter and Hosseini-Nasab, Mohammad. Theory for high-order bounds in functional principal components analysis. *Math Proc Cambridge*, 2009.

Heckel, Reinhard and Bölcskei, Helmut. Robust subspace clustering via thresholding. *arXiv preprint arXiv:1307.4891*, 2013.

Heckel, Reinhard, Tschannen, Michael, and Bölcskei, Helmut. Subspace clustering of dimensionality-reduced data. In *ISIT*, 2014.

Jacques, Julien and Preda, Cristian. Functional data clustering: a survey. *Advances in Data Analysis and Classification*, 2014.

Jaggi, Martin. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML*, 2013.

James, Gareth M. and Sugar, Catherine A. Clustering for sparsely sampled functional data. *JASA*, 2003.

Jebara, Tony, Song, Yingbo, and Thadani, Kapil. Spectral clustering and embedding with hidden markov models. In *ECML*, pp. 164–175. Springer, 2007.

Kim, Seyoung and Smyth, Padhraic. Segmental hidden markov models with random effects for waveform modeling. *JMLR*, 7:945–969, 2006.

Kriegel, Hans-Peter, Kröger, Peer, and Zimek, Arthur. Subspace clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2012.

Lin, Jessica, Keogh, Eamonn, Wei, Li, and Lonardi, Stefano. Experiencing sax: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15(2):107–144, 2007.

Lin, Jessica, Williamson, Sheri, Borne, Kirk, and DeBarr, David. Pattern recognition in time series. *Advances in Machine Learning and Data Mining for Astronomy*, 1: 617–645, 2012.

Müller, Hans-Georg. Functional data analysis. In *International Encyclopedia of Statistical Science*, pp. 554–555. Springer, 2011.

Müller, Meinard. *Information retrieval for music and motion*, volume 2. Springer, 2007.

Ng, Andrew Y, Jordan, Michael I, Weiss, Yair, et al. On spectral clustering: Analysis and an algorithm. *NIPS*, 2002.

Park, Dohyung, Caramanis, Constantine, and Sanghavi, Sujay. Greedy subspace clustering. In *NIPS*. 2014.

Paxton, Chris, Niculescu-Mizil, Alexandru, and Saria, Suchi. Developing predictive models using electronic medical records: challenges and pitfalls. In *AMIA*, 2013.

Petitjean, Francois, Forestier, Germain, Webb, Geoffrey I., Nicholson, Ann E., Chen, Yanping, and Keogh, Eamonn. Dynamic time warping averaging of time series allows faster and more accurate classification. In *ICDM*, 2014.

Rakthanmanon, Thanawin, Campana, Bilson, Mueen, Abdullah, Batista, Gustavo, Westover, Brandon, Zhu, Qiang, Zakaria, Jesin, and Keogh, Eamonn. Searching and mining trillions of time series subsequences under dynamic time warping. In *KDD*, 2012.

Sakoe, Hiroaki and Chiba, Seibi. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(1):43–49, 1978.

Saria, Suchi, Duchi, Andrew, and Koller, Daphne. Discovering deformable motifs in continuous time series data. In *IJCAI*, 2011.

Schölkopf, Bernhard and Smola, Alexander J. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

Shang, Han Lin. A survey of functional principal component analysis. *AStA Advances in Statistical Analysis*, 2013.

Silva, Ikaro, Moody, George, Scott, Daniel J, Celi, Leo A, and Mark, Roger G. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. *Computing in cardiology*, 2012.

Soltanolkotabi, Mahdi, Elhamifar, Ehsan, and Candes, Emmanuel J. Robust subspace clustering. *The Annals of Statistics*, 42(2):669–699, 2014.

Tibshirani, Robert, Walther, Guenther, and Hastie, Trevor. Estimating the number of clusters in a data set via the gap statistic. *JRSS-B*, 63(2):411–423, 2001.

Tibshirani, Robert, Saunders, Michael, Rosset, Saharon, Zhu, Ji, and Knight, Keith. Sparsity and smoothness via the fused lasso. *JRSS-B*, 2005.

Vidal, Rene. Subspace Clustering. *IEEE Signal Processing Magazine*, 2011.

Vintsyuk, T.K. Speech discrimination by dynamic programming. *Cybernetics*, 4(1):52–57, 1968.

Von Luxburg, Ulrike. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

Wang, X, Ray, S, and Mallick, BK. Bayesian curve classification using wavelets. *JASA*, 2007.

Warren Liao, T. Clustering of time series data—a survey. *Pattern recognition*, 38(11):1857–1874, 2005.

Yuan, Xiao-Tong and Li, Ping. Sparse additive subspace clustering. In *ECCV*. 2014.

A. Proof of the statement in Eq. (3)

In order to show the the result in Eq. (3), we breakdown the process in Eq. (2) into two steps: Let us denote $\widetilde{X} = \sum_{\boldsymbol{\phi}_k \in \Phi} \alpha_k \boldsymbol{\phi}_k$ and $X = d(\widetilde{X})$ where Φ is a set of s basis functions. Since the set of $\widetilde{X}$ functions create a linear subspace, every member can be written as a linear combination of at least s other functions:

$$\widetilde{X}_i = \sum_{j \neq i} \beta_j \widetilde{X}_j. \tag{7}$$

Given the fact that the set of deformations is a group, the inverse of deformation operators are also in the set and we can rewrite Eq. (7) as

$$d_i^{-1}(X_i) = \sum_{j \neq i} \beta_j d_j^{-1}(X_j), \tag{8}$$

$$X_i = d_i\Big(\sum_{j \neq i} \beta_j d_j^{-1}(X_j)\Big). \tag{9}$$

Since the operators are assumed to be linear maps, we can rewrite Eq. (9) as follows

$$X_i = \sum_{j \neq i} \beta_j d_i(d_j^{-1}(X_j)). \tag{10}$$

Group's closure property guarantees that for all i and j, there exists $\widetilde{d}_j$ in the group such that $\widetilde{d}_j = d_i \circ d_j^{-1}$. Thus we can rewrite Eq. (10) as

$$X_i = \sum_{j \neq i} \beta_j \widetilde{d}_j(X_j).$$

B. Proof of the Theorem

To prove the statement of the theorem, we need to show that by selection of the termination criterion as the theorem suggests, the Algorithm 1 will stop before adding any functions from other subspaces. In other words, Let us study the correctness of the theorem for neighbors of an arbitrary function $Y_i \in \mathcal{Y}_\ell$; heretoafter we drop the i index for simplicity of notation whenever it is not ambiguous. If R_k denotes the residual at kth step, define the normalized residual as $\bar{R}_k = R_k / \|R_k\|_2$; we need to show that the following quality cannot be larger than ϵ:

$$\max_{V \notin \mathcal{Y}_\ell, d} \left\langle \bar{R}_k, \bar{d}(V) \right\rangle < \epsilon.$$

where $\bar{d}(Y) = d(Y)/\|d(Y)\|_2$ for any function Y. Furthermore, define

$$\mu_\ell = \max_{\ell' \neq \ell} \sup_{V \in S_\ell, U \in S_{\ell'}, d, d'} \frac{|\langle d(V), d'(U) \rangle|}{\|d(V)\|_2 \|d'(U)\|_2}.$$

We note that we always have $\mu_\ell \leq \theta_\ell$, as $\mathcal{Y}_\ell \subset S_\ell$. Also, let us define the span of $d(\mathcal{Y}_\ell)$ as the span of the set of functions $\{d(Y) \big| Y \in \mathcal{Y}_\ell\}$.

To prove the main statement, we proceed with induction, as in (Dyer et al., 2013). Given the assumptions and the value of ϵ in the theorem, the first step holds, because $R_k = Y_i$. For induction, assume that at kth iteration all of the previous functions have been selected from the correct subspace. Given the result in Eq. (3), the residual is still in the span of the $d(\mathcal{Y}_\ell)$. Thus, we can write $\bar{R}_k = \bar{d}_1(U) + E$ where U is the closest function in $\mathcal{Y}_\ell$ to $\bar{R}_k$ and $E \in S_\ell$. The latter is due to the assumptions about the deformation operators that require them to be linear map and form a group with composition operation as the group law. We can write:

$$\begin{aligned}
&\max_{Y_j \notin \mathcal{Y}_\ell, d_1, d_2} |\bar{R}_k, \bar{d}_2(Y_j)\rangle| \\
&\quad = \max_{Y_j \notin \mathcal{Y}_\ell, d_1, d_2} |\langle \bar{d}_1(U) + E, \bar{d}_2(Y_j)\rangle| \\
&\quad \leq \max_{Y_j \notin \mathcal{Y}_\ell, d_1, d_2} |\langle \bar{d}_1(U), \bar{d}_2(Y_j)\rangle| + |\langle E, \bar{d}_2(Y_j)\rangle| \\
&\quad \leq \mu_\ell + \max_{Y_j \notin \mathcal{Y}_\ell, d_1, d_2} |\langle E, \bar{d}_2(Y_j)\rangle| \\
&\quad \leq \mu_\ell + \cos\theta_0 \|E\|_2 \|\bar{d}_2(Y_j)\|_2,
\end{aligned} \tag{11}$$

where θ is the minimum principal angle between S_i and all other subspaces. We can bound the $\|E\|_2$ as follows:

$$\begin{aligned}
\|E\|_2 &= \|\bar{R}_k - \bar{d}_1(U)\|_2 \\
&= \sqrt{\|\bar{R}_k\|_2^2 + \|\bar{d}_1(U)\|_2^2 - 2\langle \bar{R}_k, \bar{d}_1(U)\rangle} \\
&\leq \sqrt{2 - 2\sqrt{1 - r_\ell^2}}.
\end{aligned} \tag{12}$$

Plugging the result in Eq. (12) in Eq. (11) yields:

$$\begin{aligned}
\max_{Y_j \notin \mathcal{Y}_1, d_1, d_2} |\bar{R}_k, \bar{d}_2(Y_j)\rangle| &\leq \mu_\ell + \cos\theta_0 \sqrt{2 - 2\sqrt{1 - r_\ell^2}} \\
&\leq \mu_\ell + \sqrt{2}\cos\theta_0 r_\ell,
\end{aligned}$$

where the last step is due to the fact that $\sqrt{1 - \sqrt{1 - x^2}} \leq x$ for $x \in [0, 1]$. Given the fact that $\cos\theta$ is an upper bound for μ_ℓ, we conclude that the algorithm will not make a mistake in selection of function in its $k + 1$st step and the induction step is correct. Thus, we obtain the statement in the theorem.

C. Spectral Clustering

Note that we use the eigen-gap statistic (Line 4 in Algorithm 2 to determine the dimension of the embedding (Tibshirani et al., 2001; Von Luxburg, 2007).

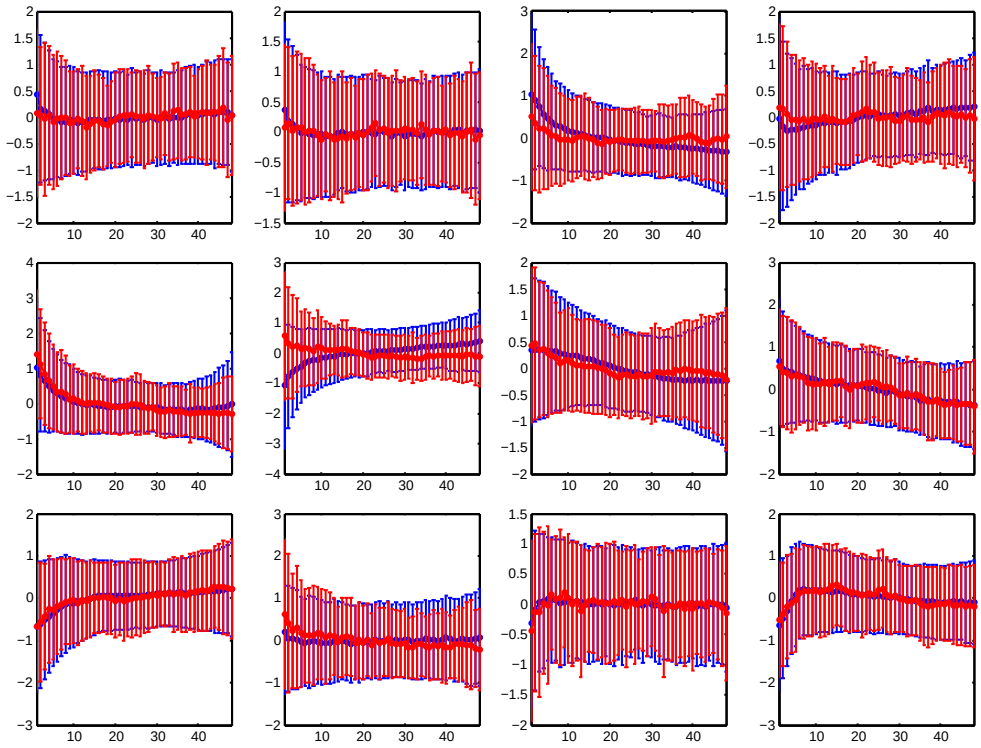


Figure 4. Mean and standard deviation trajectories for twelve variables in Physionet dataset, for patients who *survived* (blue) and *deceased* (red). Note the similarity of time series and the fact that they are almost indistinguishable by naked eye.

Algorithm 2: Spectral clustering for FSC.

Data: Affinity matrix $\boldsymbol{A}$
Result: Clustering assignments for Y_i, $i = 1, \ldots, n$.

$\boldsymbol{D} \leftarrow \text{diag}(\boldsymbol{A}\mathbf{1})$
$\boldsymbol{L} \leftarrow \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$
$\boldsymbol{\lambda}, \boldsymbol{V} \leftarrow \text{eig}(\boldsymbol{L})$
$m^\star \leftarrow \text{argmax}_{i=1,\ldots,n-1}(\lambda_i - \lambda_{i+1})$
Apply k-means to the first $m^\star$ column of $\boldsymbol{V}$.