

Clustering functional data using forward search based on functional spatial ranks with medical applications

Statistical Methods in Medical Research
2022, Vol. 31(1) 47–61
© The Author(s) 2021



Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/09622802211002865
journals.sagepub.com/home/smm



Mohammed Baragilly^{1,2} , Hend Gabr³ and Brian H Willis² 

Abstract

Cluster analysis of functional data is finding increasing application in the field of medical research and statistics. Here we introduce a functional version of the forward search methodology for the purpose of functional data clustering. The proposed forward search algorithm is based on the functional spatial ranks and is a data-driven non-parametric method. It does not require any preprocessing functional data steps, nor does it require any dimension reduction before clustering. The Forward Search Based on Functional Spatial Rank (FSFSR) algorithm identifies the number of clusters in the curves and provides the basis for the accurate assignment of each curve to its cluster. We apply it to three simulated datasets and two real medical datasets, and compare it with six other standard methods. Based on both simulated and real data, the FSFSR algorithm identifies the correct number of clusters. Furthermore, when compared with six standard methods used for clustering and classification, it records the lowest misclassification rate. We conclude that the FSFSR algorithm has the potential to cluster and classify functional data.

Keywords

Cluster analysis, forward search, functional data, nonparametric methods, spatial ranks

1 Introduction

In many medical applications, the observed data may be assumed to have arisen from a continuous curve or higher dimensional surface that is described by some function. Thus, the glucose levels as measured by a continuous glucose monitor or the tracings generated by an electroencephalogram of a patient are both examples where a continuous function may be used to help describe the underlying data.

In both of these examples, a single function is assumed to underlie the data for an individual patient so that for a sample of patients there is a sample of functions in which the data are observed at particular points. In this regard, the function is the element of interest¹ and analysis of the shape of these functions can help inform decisions on classification and prediction.^{2,3}

Functional data analysis describes the statistical methods and techniques that are used to explore functional data.¹ The random variable is a functional, that is, a space of functions, defined on some continuous interval such as time.^{1,2} Thus, each realization of the variable is a function providing infinitely dimensional data, and the space of functions is generally assumed to be a Hilbert space.⁴ Although our concern here will be with univariate functional data, sometimes multivariate functional data may be of interest.^{1,5,6} In practice, whichever the type of data, the functions are often sampled at a finite set of points.

¹Department of Mathematics, Insurance and Applied Statistics, Helwan University, Helwan, Egypt

²Institute of Applied Health Research, University of Birmingham, Birmingham, UK

³Faculty of Commerce, Menoufia University, Al Minufya, Egypt

Corresponding author:

Brian Harvey Willis, Institute of Applied Health Research, University of Birmingham, Edgbaston, Birmingham B15 2TT, UK.
Email: b.h.willis@bham.ac.uk

In many situations, we need to know the hidden structure that explains how these curves and functions vary from one group to another. Thus, in the study of childhood obesity, growth curves of body measurements may be used to group children using a cluster method.⁷ One such cluster method, k -means clustering, has been used to identify patterns of multi-morbidity and check whether these patterns are stable over time for a cohort of older people.⁸

For many of the cluster methods used on functional data, the number of clusters is assumed to be fixed a priori. This makes determining the optimal number of clusters in the functional data important and motivates this research. Thus, the method proposed here can be used to identify the number of clusters and is a development on the forward search originally used to identify outliers in multivariate data^{9,10} and later, as a clustering method.¹¹

Here we use a forward search based on functional spatial ranks to analyze functional data. This extends previous work that introduced the forward search based on spatial ranks for the cluster analysis of multivariate data.¹² The functional forward search introduced here is based on the random start forward search,¹³ and can be considered a new raw-data method that obviates the need for dimension reduction, since it performs the clustering directly on the discrete observation of the curves or functions.

It is a non-parametric method that can be used to determine the number of clusters, and assign each curve to its cluster. When compared with existing methods using different numerical examples from real data, it is shown to be an effective tool in clustering analysis.

The paper is organized as follows. In section 2, we discuss the curse of dimensionality in the traditional random start forward search method and the potential of using the forward search based on functional spatial ranks. In section 3, we propose the functional forward search algorithm based on functional spatial ranks. In section 4, we compare the proposed method with other functional data clustering methods using numerical examples before ending with the discussion in section 5.

2 The curse of dimensionality in the traditional forward search

The term “curse of dimensionality” was introduced by Bellman.¹⁴ It refers to all problems caused by the analysis of high-dimensional data and, in general, arises from a relative sparsity of observations. For example, in order to run the traditional forward search algorithm based on Mahalanobis distances, we need to choose an initial subset $S(m)$, with $m = d + 1$ for some dimension, d . Practically, this is not difficult if the number of available observations is large compared to the number of variables, although the traditional forward search algorithm becomes less efficient as the dimension d grows.

In contrast, when the number of observations is small compared to the number of variables it is not possible to estimate the variance–covariance matrix so the algorithm cannot proceed. Strictly, the traditional forward search based on Mahalanobis distances¹¹ cannot be applied to functional data owing to the random variables taking values into an infinite dimensional space. However, in practice the data consist of curves that have been sampled at a finite set of points, hence it is still possible to use forward search methods providing the dimension, d is less than the underlying sample size.

The forward search, like all multivariate methods based on Mahalanobis distances, suffers when the dimension grows. Since it starts with subsets of size $d + 1$, it is unable to identify clusters of a size less than d . Thus, when both the number of variables is very big and some of the clusters are of a small size, this algorithm will lead to information loss about the number of clusters to be determined.

In contrast, using a forward search based on spatial ranks for clustering multivariate data overcomes this issue,¹² as it can be started with subsets of any size since the rank of any observation $\mathbf{x} \in \mathbb{R}^d$ with respect to a single data point is always 1.¹² Similarly, we can use the non-parametric forward search based on the functional spatial ranks for clustering functional data, even though functional data is intrinsically different from multivariate data. This algorithm is described in more detail in the next section.

3 The functional forward search based on functional spatial ranks

In this section, we propose a functional forward search algorithm based on functional spatial ranks. Part of the novelty of this algorithm is that unlike the traditional forward search algorithm, it works with functional data. Furthermore, as a raw data method it determines the number of clusters from the data without any need for parameter estimation. A key element is the need to extend $sign(\mathbf{x})$ and $Rank(\mathbf{x})$ naturally from \mathbb{R}^d to any

infinite-dimensional Hilbert space \mathbb{H} . We start with a review of the relevant literature before defining the functional sign and spatial ranks function.

3.1 The functional spatial rank

A spatial approach to multivariate and functional data appeared as early as 1983, when the spatial median was used for robust location estimation for two dimensional spatial data.¹⁵ The development of non-parametric geometrical approaches led to the introduction of multivariate spatial quantiles¹⁶ and the multivariate spatial depth function.¹⁷

The functional spatial depth (FSD), proposed by Chakraborty and Chaudhuri,¹⁸ extends the notion of spatial depth from d -dimensional multivariate space \mathbb{R}^d into infinite dimensional spaces. As a result, the multivariate spatial depth function, $SD(\mathbf{x}) = 1 - \|\mathbb{E}\{(\mathbf{x} - \mathbf{X})/\|\mathbf{x} - \mathbf{X}\|\}\|$, where $SD(\mathbf{x})$ is the spatial depth of $\mathbf{x} \in \mathbb{R}^d$ with respect to the probability distribution of a random vector $\mathbf{X} \in \mathbb{R}^d$, can be extended naturally to any Hilbert space \mathbb{H} . Thus, for any $\mathbf{x} \in \mathbb{H}$ and a random element $\mathbf{X} \in \mathbb{H}$, the $FSD(\mathbf{x})$ is defined based on same expression as SD , where $\|\cdot\|$ is the usual norm in \mathbb{H} and the expectation \mathbb{E} is estimated based on the Bochner integral.^{18,19}

The spatial depth function has been used to provide a nonparametric description of functional data, by using the functional version of spatial depth to identify some nonparametric descriptive features such as sample median and quantile curves.²⁰

The functional spatial median has been of particular interest to investigators. For example, Cardot et al.²¹ used an averaged stochastic gradient algorithm to compute the functional spatial median in a Hilbert space in a fast way. And this functional spatial median has been used as a robust measure of center for a data set of electricity loading curves.²² The kernelized functional spatial depth (KFSD) has been proposed²³ for the classification of functional data. It is based on the functional spatial depth introduced by Serfling and Wijesuriya.¹⁷ In addition, the functional K-nearest neighbour classifier has been used in this work as a benchmark procedure.

Suppose that X is a random variable with values in an infinite dimensional space or functional space. For instance, the stochastic process $X = \{X(t); t \in \mathcal{T}\}$; where $\mathcal{T} \subset \mathbb{R}$ is a good example for the functional variable X , which takes values in some Hilbert space \mathbb{H} of functions defined on some set \mathcal{T} , where \mathcal{T} represents an interval of time, of wavelengths or any other subset of \mathbb{R} .¹ We now define the functional spatial rank. Suppose that $\{\mathbf{X}_1(t), \mathbf{X}_2(t), \dots, \mathbf{X}_n(t)\}$ is a functional dataset based on the functional random variables $\{X_1(t), X_2(t), \dots, X_n(t)\}$ that take values in \mathbb{H} , and t is defined on some continuous interval \mathcal{T} , then the population functional spatial rank function for the curve $\mathbf{x}(t) \in \mathbb{H}$ is defined as

$$FSR_F(\mathbf{x}(t)) = \mathbb{E} \left[\frac{\mathbf{x}(t) - \mathbf{X}(t)}{\|\mathbf{x}(t) - \mathbf{X}(t)\|} \right] \quad (1)$$

where $\|\mathbf{x}(t)\|$ is the l_2 norm

$$\|\mathbf{x}(t)\| = \left\{ \int_{\mathcal{T}} \{x(t)\}^2 dt \right\}^{\frac{1}{2}} \quad (2)$$

for an infinite dimensional space.

In practice, the curves are observed at a finite set of points, so that there are discrete observations for each sample path $\mathbf{X}_i(t)$ at a finite set of knots $\{t_{ij} : j = 1, \dots, m_i\}$. Thus we have $\mathbf{X}_i(t) = \{\mathbf{X}(t_{i1}), \mathbf{X}(t_{i2}), \dots, \mathbf{X}(t_{imi})\}$, and for shorthand $\mathbf{X}_{ij} = \mathbf{X}(t_{ij})$. Here, we consider regularly sampled curves, where the evaluation points $t \in \mathcal{T}$ are fixed for each curve, with the same length and knots, so that $\{t_{ij} : j = 1, \dots, m\}$ and $i = 1, \dots, n$. The corresponding l_2 norm of FSR_{F_n} denoted by $FSRN_{F_n}$ is given by $\|FSR_{F_n}(\mathbf{x}(t))\|$. Thus, if $\{\mathbf{X}_1(t), \mathbf{X}_2(t), \dots, \mathbf{X}_n(t)\}$ is the set of curves regularly sampled at a finite set of observations, then the sample functional spatial rank of $\mathbf{x}(t)$ with respect to $\mathbf{X}_1(t), \mathbf{X}_2(t), \dots, \mathbf{X}_n(t)$ is given by

$$FSR_{F_n}(\mathbf{x}(t)) = \frac{1}{n} \sum_{i=1}^n \frac{\mathbf{x}(t) - \mathbf{X}_i(t)}{\|\mathbf{x}(t) - \mathbf{X}_i(t)\|} \quad (3)$$

As a vector, the functional spatial rank provides information on the centrality of an observed curve and its direction. The l_2 norm $\|FSR_{F_n}(\mathbf{x}(t))\|$, which is bounded to lie in the interval $[0, 1)$, provides a measure of “distance” of $FSR_{F_n}(\mathbf{x}(t))$ from the spatial median of the functional data. Thus, when the $\|FSR_{F_n}(\mathbf{x}(t))\|$ is close to zero, the $\mathbf{x}(t)$ will be close to the spatial median. In contrast, if $\|FSR_{F_n}(\mathbf{x}(t))\|$ is close to one, the $\mathbf{x}(t)$ could be an outlier curve and potentially provides the basis for an approach to outlier detection.

Clearly it requires deciding upon a suitable cut-off for an outlier and one approach is to trim the sample of a proportion of curves with the highest $FSRN_{F_i}(\mathbf{x}(t))$. Thus trimming the sample of the top 1%, 2%, 5%, and 10%, has been used to investigate the stability of the clusters when a parametric model-based clustering approach has been used.^{24,25}

A simpler approach is to derive the cut-off, C based on the upper whisker of the boxplot of $FSRN_{F_i}(\mathbf{x}(t))$ using the formula $C = Q_3 + (1.5 \times IQR)$ where Q_3 is the upper quartile and IQR is the interquartile range ($Q_3 - Q_1$) when Q_1 is the lower quartile. Those curves with $FSRN_{F_i}(\mathbf{x}(t))$ exceeding C are then considered outliers and this is the approach used here.

In principle, the functional spatial ranks can be applied for both regularly and irregularly sampled curves, where the functional spatial ranks are supposed to be calculated in general concept using the integrations instead of the summations quantities, and then with a formal procedure and methods we can estimate the integral functions and get the estimated values of the functional spatial ranks. Alternatively, we may use some smoothing functions or spline coefficients to get an equal length of the irregularly sampled curves, and then we can use the above equations to obtain the functional spatial ranks of the irregularly sampled curves.

3.2 Functional spatial ranks classifier

Before introducing the forward search algorithm, we consider the problem of classifying functional data to particular clusters. In general, it is important to assess whether the curves have been appropriately assigned to a cluster and whether they remain unassigned to any cluster. A further problem that may arise with some algorithms classifying functional data is when some curves are assigned to more than one cluster.

Clearly it is desirable to have a mechanism of assigning each curve in the functional data to an appropriate cluster. Here we use a nonparametric classifier based on the functional spatial ranks that is applied after determining the number of clusters. Assuming we have k groups of observations, with population distributions F_1, F_2, \dots, F_k , we may assign $\mathbf{x}(t)$ to the group in which the l_2 norm of the functional spatial ranks based on F_i is smallest such that

$$FSRN_{F_i}(\mathbf{x}(t)) = \min_{1 \leq j \leq k} FSRN_{F_j}(\mathbf{x}(t)) \quad (4)$$

where $i \neq j, 1 \leq i \leq k$. Thus, the forward search algorithm that follows identifies the number of clusters then applies the classifier in equation (4) to assign each curve to the most suitable cluster.

3.3 The forward search based on functional spatial rank algorithm

Let $S(m)$ be a subset from the observed curves of size m . Define the functional spatial ranks of an individual curve corresponding to the subset $S(m)$ as

$$r_i(m) = \frac{1}{m} \sum_{j \in S(m)} \frac{\mathbf{X}_i(t) - \mathbf{X}_j(t)}{\|\mathbf{X}_i(t) - \mathbf{X}_j(t)\|} \quad (5)$$

where $i = 1, \dots, n$. The functional forward search algorithm with functional spatial ranks (FSFSR) is as follows:

1. Selecting random starting points, the search is started with an initial subset $S(m)$ with size $m = 3$.
2. Calculate the functional spatial ranks $r_i(m)$ of the curves in the subset $S(m)$.
3. Compute $r_{min}(m)$, where $r_{min}(m) = \min \|r_i(m)\|; i \notin S(m)$, where $\|\cdot\|$ is the Euclidean norm, such that $\|r_i(m)\| = \sqrt{r_i(m)_1^2 + r_i(m)_2^2 + \dots + r_i(m)_t^2}$.
4. Grow the subset $S(m)$ to $S(m+1)$ by taking $m+1$ curves $\mathbf{X}_i(t)$'s, which correspond to the smallest $m+1$ norms $\|r_i(m)\|$'s, where $\|\cdot\|$ is the Euclidean norm as defined in step 3. Set $m = m+1$.

5. Repeat 2 – 4 until $m = n - 1$.
6. Plot $r_{\min}(m)$ against the corresponding subset sizes, m to get the forward plot and identify the number of clusters.
7. Identify the subset size by finding the highest $r_{\min}(m)$ around each peak and set m as the cluster size. To specify the membership of each group, we may stop the search at each peak and set the curves included in m as the cluster's membership.
8. Apply the functional spatial ranks classifier in section 3.2 to confirm the assignment of each curve and allocate the unassigned/incorporated curves to the proper group.

Theoretically, the algorithm can be started with any number of random starting points, since it is not constrained with the rule of $m = d + 1$ as in the multivariate case. However, using a small number of the random starting points improves the chances of detecting small clusters in the data; hence we start the search with $m = 3$. The trade-off is that the search may produce a large number of peaks for small values of m and it may be difficult to determine number of clusters visually and their respective sizes.

The computation of $r_i(m)$ may be computationally expensive in very high dimensions and very large samples sizes. As a result, increasing the number of random start forward searches increases the computational time. In order to elucidate the structure of the data within a reasonable computational time, we use 100 randomly chosen initial subsets. However, for lower dimensions and small sample sizes, we may increase the number of random start forward searches to 200 or 300. This reduces the risk of missing a trajectory with a different pattern.

When the curves in $S(m)$ belong to the same cluster, the $\|r_i(m)\|$ for a curve $\mathbf{X}_i(t)$ in that cluster is expected to be smaller than that for a curve from a different cluster. Furthermore, as $S(m)$ grows, we expect to see a jump in the magnitude of the rank function when the nearest point to $S(m)$ is from a different cluster. So, we may determine the number of clusters and their sizes in the functional data using the forward plot based on the functional spatial ranks.

4 Numerical examples

In this section, we apply the FSFSR algorithm proposed in section 3.2 to some numerical examples. The first three examples are simulated data generated from three different models. The final two examples use data from real datasets.

To assess the performance of the FSFSR algorithm, it is important to recognize that the algorithm both identifies the number of clusters and assigns all the data to an appropriate cluster. Thus, any performance metric must capture both of these elements and penalize the performance when either it identifies an incorrect number of clusters or wrongly assigns data to a cluster. Thus, we use the following misclassification rate, which is similar to the classification error proposed by Meila.²⁶

For n data points, suppose there are r true classes $T = \{T_1, T_2, \dots, T_r\}$, and k clusters based on the clustering algorithm $C = \{C_1, C_2, \dots, C_k\}$. And define the two vectors A and B such that $A = \{1, 2, \dots, k\}$ and $B = \{1, 2, \dots, r\}$. Then the misclassification rate, H can be defined as

$$H = 1 - \left(\frac{1}{n}\right) \max \left(\sum_{(i,j) \in A \times B} C_i \cap T_j \right) \quad (6)$$

with the condition that if the two terms $C_i \cap T_j$ and $C_t \cap T_u$ appear in the sum then $i = t$ if and only if $j = u$. This guarantees the rows and columns of the matrix $A \times B$ contribute at most one element to the summation. Consequently, the term $C_i \cap T_u$ is set to zero if the term $C_i \cap T_j$ is one of the terms that maximizes the sum in parentheses. For $k = 1$, the sum would contain only one term.

The adjusted Rand Index (ARI) is also popular metric used for measuring the performance of clustering algorithms and it has been included here for completeness.²⁷ In contrast to H , which compares clusters by matching sets, the ARI compares clusters by counting the pairs of points in which the clusters agree or disagree. It also corrects for the expected value of the unadjusted Rand Index, where the expected value is based on a random choice of entries in the contingency table when the column and row totals are fixed.

For comparison, we considered six other methods for identifying the number of clusters and cluster sizes. Note in what follows the term in brackets, after the method name, corresponds to the function and package in **R**. The first method is model-based clustering (*mclust*)²⁸ based on a Gaussian mixture model (GMM).²⁹ The number of

clusters is determined by the model which returns the largest Bayesian information criterion (BIC). For the second method, the *K-means*,³⁰ the number of clusters needs to be set in advance, and here it corresponds to the number which returns the largest CH index.³¹ The third method is the high dimensional data clustering method (*HDCC*).³² It is a model-based clustering method also based on the GMM, where the number of clusters corresponds to the model which returns the largest BIC. The fourth method is the mixtures of probabilistic principle component analyzers (*MixtPPCA*)³³ which again uses the model with the largest BIC to determine the number of clusters. The partitioning around medoids (*PAM*)³⁴ is the fifth method considered. Here the number of clusters is selected based on the optimum average silhouette width.³⁵ Finally, the sixth method is the functional high-dimensional data clustering method (*FunHDCC*) which is an adaptive method that uses the functional data directly and chooses the number of clusters based on the largest BIC value.³⁶

The first five methods were implemented as a raw-data method with discretized data and as a filtering method based on 10 spline coefficients. All six methods were applied to the three simulated and two real datasets and compared with the FSFSR algorithm using the number of clusters identified, H^{26} and ARI.²⁷

For the *K-means*³⁰ and *HDCC*³² methods the initial partitioning of the data points is random, which may result in the values of the performance measures varying between different runs of the algorithm. For these cases, the algorithms were repeated 1000 times and the average over the repetitions has been calculated.

4.1 Simulated data examples

The first simulated data (model 1) consists of two groups. The first group includes curves that are generated from the process

$$X(t) = m_0(t) + e(t) \quad (7)$$

with mean function $m_0(t) = -35(1-t)t^{1.4}$ and $e(t)$ is a Gaussian process with mean 0 and $Cov(X(s), X(t)) = 0.3\exp(-s-t/0.4)$. Here, t is a sequence of numbers between 0 and 1 with length 100. The second group consists of the generated curves from the process

$$Y(t) = m_1(t) + e(t) \quad (8)$$

where the mean function $m_1(t) = -35t(1-t)^{1.4}$, with both Gaussian process $e(t)$ and $Cov(X(s), X(t))$ are defined as in $X(t)$ in equation (6). So, the mixture model $Z(t)$ consists of the two groups $X(t)$ and $Y(t)$ such that, $Z(t) = pY(t) + (1-p)X(t)$, where p is the mixing proportion, which is the probability of an individual curve being generated by specific process.

A mixing proportion of 0.25 was used to generate the two clusters with a sample size $n=160$ curves. The black curves, in Figures 1(a), represent the first group, and the red curves represent the second. In Figure 1(b), the mean function of model 1 is given. As the sample size is 160 and the mixing proportion is 0.25, the first cluster includes 40 curves and the second cluster includes 120 curves. Panel (c) of Figure 1 gives the forward plot of the simulated functional data based on the functional spatial ranks. It can be seen from panel (c), there are two maxima at sizes $m=38$ and 120, which suggests the data have been correctly divided into two groups. To identify the membership of the two clusters, we stopped the algorithm at $m=38$ and $m=120$. Before applying the classifier in step 7 of the algorithm, two curves remained unassigned. These were assigned to each cluster after applying the classifier resulting in two clusters of size 39 and 121, respectively. Comparing the clusters label with the simulated classes label, 159 curves out of 160 have been assigned correctly. This gives an H of 0.00625, and an ARI of 0.973.

In Table 1 the performances of all the algorithms are summarized. Many of the algorithms identified the correct number of clusters and achieved perfect classification. However, three algorithms (*mclust*, *HDCC* and *MixtPPCA*) when implemented as raw-data methods with discretized data returned an incorrect number of clusters. This affected their respective misclassification rates and adjusted Rand indices.

In the second model, there are also two groups. The first group consists of curves generated from the process similar to equation (7) but with a different mean function: $m_0(t) = -35(1-t)t^3 + |4\sin(25\pi t)|$ and $e(t)$ is the same Gaussian process as defined in process (7). The second group is a smoothing of the curves of the first group, and it consists of spline approximations (with eight knots) of the trajectories in the first group. This in fact makes it more difficult to discriminate the overlap between the two clusters. Similarly, we set $n=160$ and the mixing proportion to 0.25.

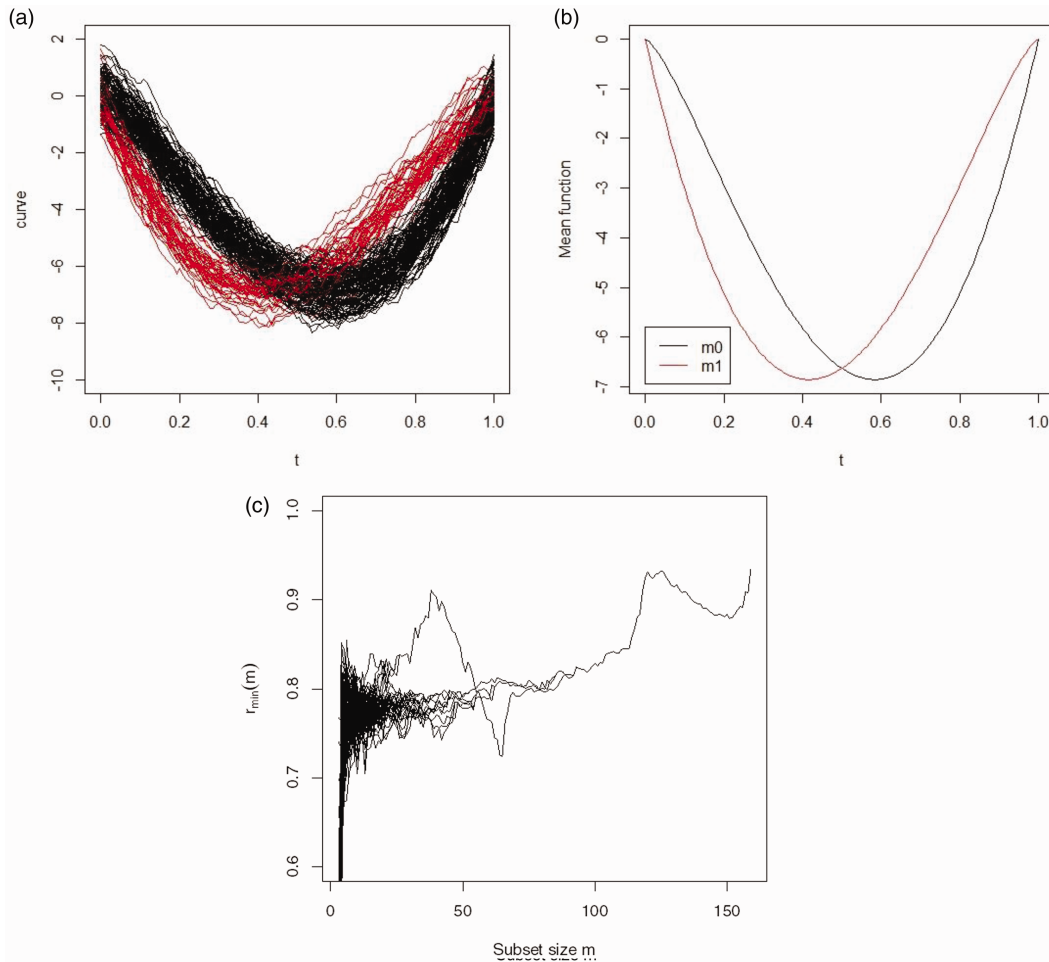


Figure 1. Simulated data, Model I: (a) the observed curves with two groups, (b) the mean function, (c) the forward plot based on functional spatial ranks.

Table 1. Comparison of different clustering approaches applied to Model I.

Clustering method	No. of clusters	Cluster sizes	H	ARI
FSFSR (a)	2	39, 121	0.00625	0.97331
GMM (mclust) "BIC" (a)	1	160	0.25000	0.00000
GMM (mclust) "BIC" (b)	2	40, 120	0.00000	1.00000
Kmeans based on CH index (a) ^a	2	40, 120	0.00000	1.00000
Kmeans based on CH index (b) ^a	2	40, 120	0.00000	1.00000
HDDC "BIC" (a) ^a	4	10, 26, 38, 86	0.24141	0.53795
HDDC "BIC" (b) ^a	2	40, 120	0.00019	0.99973
MixtPPCA "BIC" (a)	1	160	0.25000	0.00000
MixtPPCA "BIC" (b)	2	40, 120	0.00000	1.00000
PAM "Silhouette width" (a)	2	40, 120	0.00000	1.00000
PAM "Silhouette width" (b)	2	40, 120	0.00000	1.00000
FunHDDC "BIC"	2	39, 121	0.00625	0.97331

Note: When a method is followed by letter in parentheses it denotes the following: (a) = raw-data methods with discretized data; (b) = filtering methods using spline coefficients from 10 splines.

^aResults are based on the mean of 1000 repetitions.

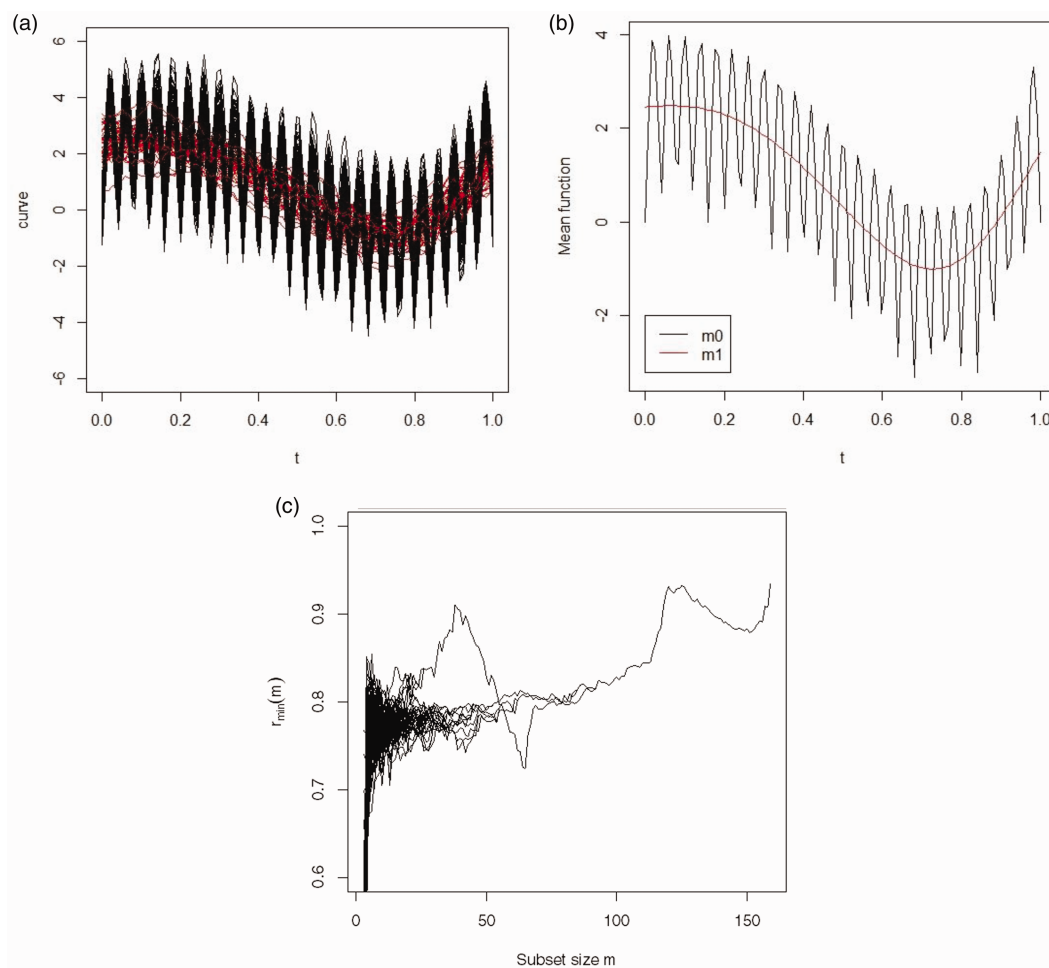


Figure 2. Simulated data, Model 2: (a) the observed curves with two groups, (b) the mean function, (c) the forward plot based on functional spatial ranks.

Table 2. Comparison of different clustering approaches applied to Model 2.

Clustering method	No. of clusters	Cluster sizes	H	ARI
FSFSR (a)	2	40, 120	0.00000	1.00000
GMM (mclust) "BIC" (a)	2	40, 120	0.00000	1.00000
GMM (mclust) "BIC" (b)	1	160	0.25000	0.00000
Kmeans based on CH index (a) ^a	2	40, 120	0.00000	1.00000
Kmeans based on CH index (b) ^a	2	79, 81	0.39443	0.04004
HDDC "BIC" (a) ^a	5	9, 19, 30, 40, 62	0.29649	0.43703
HDDC "BIC" (b) ^a	1	160	0.25000	0.00000
MixtPPCA "BIC" (a)	1	160	0.25000	0.00000
MixtPPCA "BIC" (b)	1	160	0.25000	0.00000
PAM "Silhouette width" (a)	2	40, 120	0.00000	1.00000
PAM "Silhouette width" (b)	2	63, 97	0.40625	0.01967
FunHDHC "BIC"	2	29, 131	0.31875	0.03015

Note: When a method is followed by letter in parentheses it denotes the following: (a) = raw-data methods with discretized data; (b) = filtering methods using spline coefficients from 10 splines.

^aResults are based on the mean of 1000 repetitions.

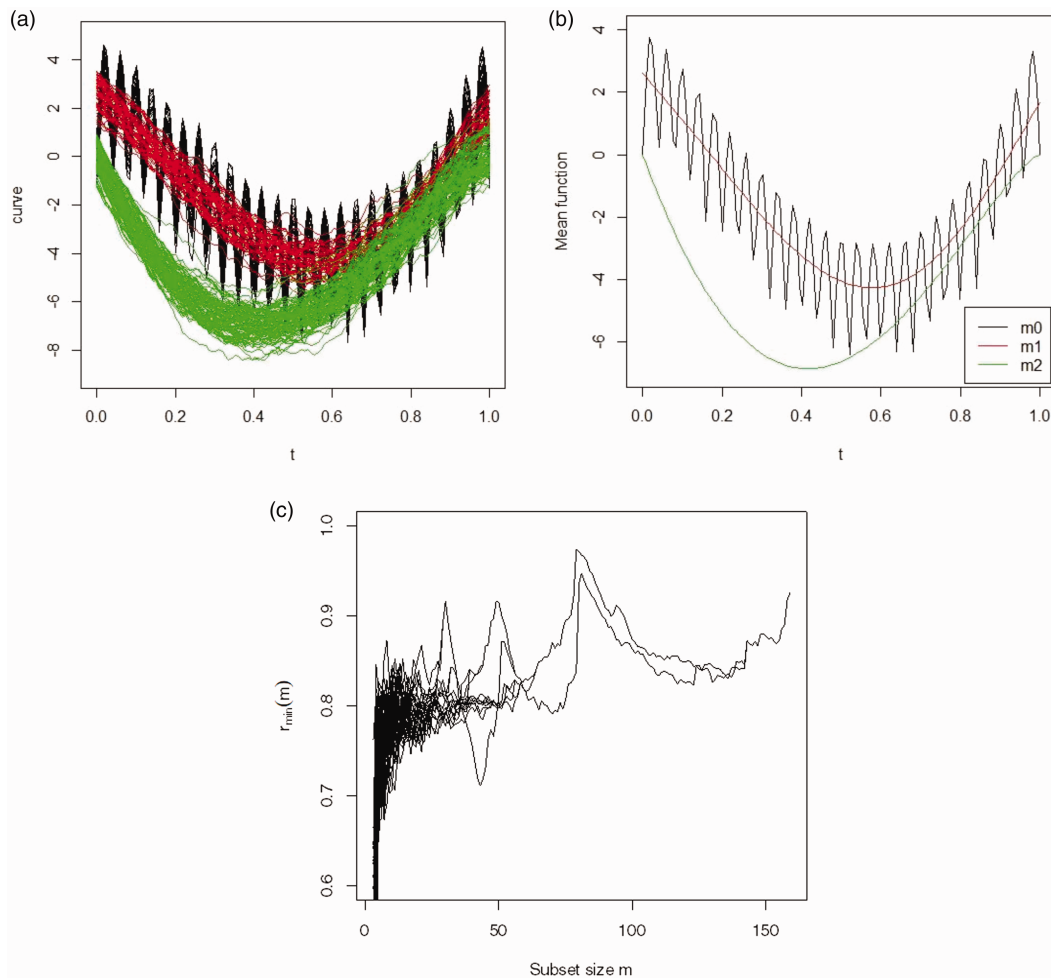


Figure 3. Simulated data, Model 3: (a) the observed curves with three groups, (b) the mean function, (c) the forward plot based on functional spatial ranks.

Figure 2(a) shows the simulated curves from the first cluster (black) and second cluster (red). The corresponding mean functions are shown in Figure 2(b). The forward plot based on the functional spatial ranks is shown in Figure 2(c). Again we can clearly see two peaks around $m = 40$ and 117 , and three curves (81, 120 and 148) have not been assigned to a cluster before the classification step. Applying the classifier assigns all three curves to S ($m = 117$), thus resulting in two clusters of sizes 40 and 120. Furthermore, all 160 curves have been classified correctly so H is 0 and the ARI is 1. In Table 2 it can be seen that including FSFSR 7/12 algorithms identified the correct number of clusters. In 3/7 which identified the correct number of clusters, H was above 0.31 and the ARIs were between 0.02 and 0.04.

For the third model, we combine the two previous models so there are three clusters. The first cluster consists of the generated curves from the process defined in equation (7) but with a different mean function: $m_0(t) = -35(1-t)t^{1.4} + |4\sin(25\pi t)|$, with both the Gaussian process $e(t)$ and $Cov(X(s), X(t))$ defined as in $X(t)$ in equation (6). The second cluster is a smoothing of the curves of the first cluster, and it is made of spline approximations (with eight knots) of the trajectories in the first cluster. The third cluster is derived from equation (8) in model 1.

The three simulated clusters have sizes 30, 50 and 80. Figure 3(a) and (b) shows the respective simulated curves for the clusters and their mean functions. In Figure 3(c), the forward plot again demonstrates that the functional forward search algorithm has identified the correct number of clusters. There are three peaks at $m = 30, 49$ and 79 . In addition, there are two unassigned curves and the classifier in step 7 assigns both of them to cluster $S(m = 49)$ giving three clusters of sizes to 30, 51 and 79. Out of 160 curves in the sample, 159 curves have been assigned correctly, resulting in an H of 0.00625 and ARI of 0.978.

Table 3. Comparison of different clustering approaches applied to Model 3.

Clustering method	No. of clusters	Cluster sizes	H	ARI
FSFSR (a)	3	30, 51, 79	0.00625	0.97843
GMM (mclust) "BIC" (a)	1	160	0.50000	0.00000
GMM (mclust) "BIC" (b)	2	80, 80	0.68750	0.76379
Kmeans based on CH index (a) ^a	2	80, 80	0.68750	0.76379
Kmeans based on CH index (b) ^a	2	80, 80	0.68750	0.76379
HDDC "BIC" (a) ^a	8	7, 12, 13, 14, 16, 22, 28, 48	0.67942	0.54742
HDDC "BIC" (b) ^a	2	80, 80	0.68750	0.76379
MixtPPCA "BIC" (a)	1	160	0.50000	0.00000
MixtPPCA "BIC" (b)	2	80, 80	0.68750	0.76379
PAM "Silhouette width" (a)	2	80, 80	0.68750	0.76379
PAM "Silhouette width" (b)	2	80, 80	0.68750	0.76379
FunHDDC "BIC"	2	80, 80	0.68750	0.76379

Note: When a method is followed by letter in parentheses it denotes the following: (a) = raw-data methods with discretized data; (b) = filtering methods using spline coefficients from 10 splines.

^aResults are based on the mean of 1000 repetitions.

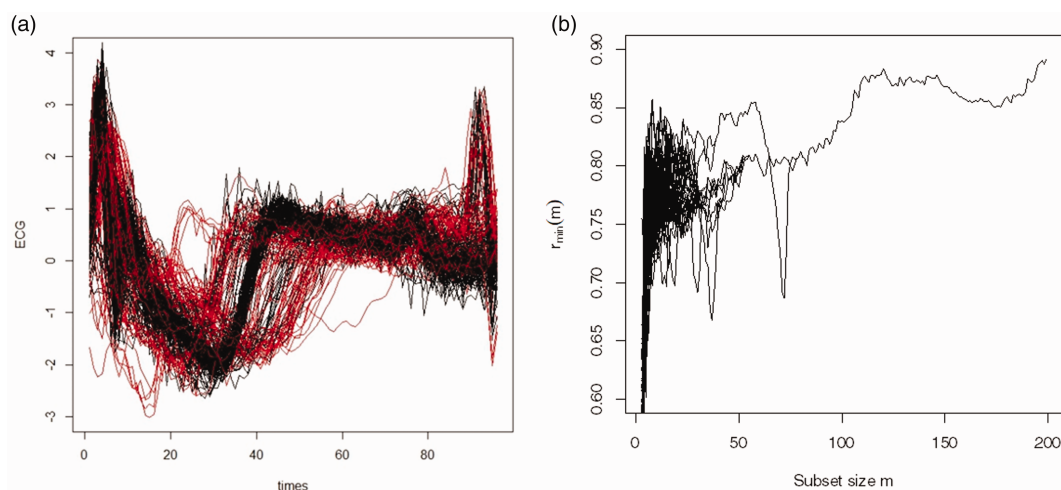


Figure 4. ECG data: panel (a) is the observed curves with two groups and panel (b) is the forward plot based on the functional spatial ranks. Two clusters are evident at subsets with sizes 58 and 120.

A comparison with the other algorithms is provided in Table 3. It can be seen that the FSFSR algorithm is the only algorithm which identifies the correct number of clusters with near perfect classification. In contrast, the other algorithms suffer significant misclassification rates although the adjusted Rand indices are less affected with many returning a value of 0.764.

4.2 Real data examples

In this section we apply the FSFSR algorithm to two real datasets. The first dataset is known as the ECG data and is taken from the UCR Time Series Classification and Clustering Archive.³⁷ The dataset consists of 200 electrocardiograms from two groups of patients sampled at 96 time points, in which 133 are classified as normal and 67 as abnormal. The data consist of the ECG signals recorded between two electrodes during one heartbeat. The abnormal ECGs reflect a cardiac pathology known as a supraventricular premature beat.

The second dataset, known as the 'DistalPhalanxOutlineCorrect' data (hereon referred to as the Distal data) is also taken from the UCR Time Series Classification and Clustering Archive.^{38,39} It is designed to test the efficacy of hand and bone outline detection by an image processing algorithm. The outlines of the three bones of the middle finger in each image are summarized by a univariate series of 80 data points representing Euclidean distances of different points around the outline from a central point. Here, we consider the test sample of 276

Table 4. Comparison of different clustering approaches applied to the ECG dataset.

Clustering method	No. of clusters	Cluster sizes	H	ARI
FSFSR (a)	2	58, 142	0.23500	0.26405
GMM (mclust) "BIC" (a)	1	200	0.33500	0.00000
GMM (mclust) "BIC" (b)	3	97, 41, 62	0.31500	0.37860
Kmeans based on CH index (a) ^a	3	111, 35, 54	0.32489	0.31411
Kmeans based on CH index (b) ^a	3	117, 48, 35	0.32000	0.32874
HDDC "BIC" (a) ^a	3	36, 51, 113	0.36217	0.32400
HDDC "BIC" (b) ^a	5	58, 14, 20, 17, 91	0.39474	0.37243
MixtPPCA "BIC" (a)	3	104, 61, 35	0.30000	0.36854
MixtPPCA "BIC" (b)	4	97, 39, 26, 38	0.37000	0.38710
PAM "Silhouette width" (a)	3	109, 55, 36	0.32000	0.33304
PAM "Silhouette width" (b)	4	118, 47, 19, 16	0.32500	0.33866
FunHDDC "BIC"	1	200	0.33500	0.00000

Note: When a method is followed by letter in parentheses it denotes the following: (a) = raw-data methods with discretized data; (b) = filtering methods using spline coefficients from 10 splines.

^aResults are based on the mean of 1000 repetitions.

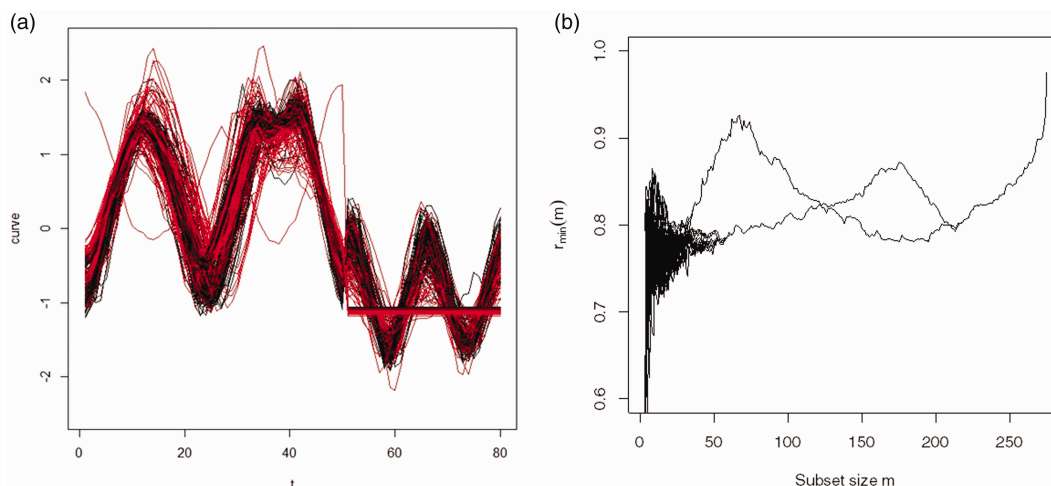


Figure 5. Distal data: panel (a) is the observed curves with two groups and panel (b) is the forward plot based on the functional spatial ranks. Two clusters are evident at subsets with sizes 67 and 175.

images. There are two classes based on whether the bones have been correctly delineated by the image processing algorithm (115) or not (161) as determined by human evaluation.

Figure 4(a) shows the observed curves for the ECG data, and the forward plot based on functional spatial rank. From Figure 4(b), two clusters are evident with peaks at 58 and the other at 120. This suggests that some of the observations have not been captured by either cluster. In order to identify the membership of each cluster, the forward search was stopped at the first peak ($m=58$) to identify the subset $S(m=58)$. Similarly, stopping the search at $m=120$ identifies the second cluster $S(m=120)$. Before applying the classifier in step 7 of the algorithm, 15 curves have been incorporated in both clusters and 37 curves have not been assigned to any cluster. Applying the functional spatial ranks classifier to these 52 curves classifies each curve to a unique cluster. As a result, H is 0.235 and the ARI is 0.264.

Table 4 gives the results for all the methods applied to the ECG data. It is clear that only the FSFSR algorithm gives the correct number of clusters (2) and has the lowest H (23.5%). Despite identifying an incorrect number of clusters, for many of the other methods the ARI is more favourable than the FSFSR algorithm. None is above 0.39 and this would suggest poor classification by all the algorithms; however, interpretation of the ARI is not straightforward as the baseline expected value for the Rand index varies as the contingency table varies.^{40,41}

Table 5. Comparison of different clustering approaches applied to the Distal dataset.

Clustering method	No. of clusters	Cluster sizes	H	ARI
FSFSR (a)	2	211, 65	0.23551	0.00512
GMM (mclust) "BIC" (a)	1	276	0.41667	0.00000
GMM (mclust) "BIC" (b)	4	140, 14, 53, 69	0.49275	0.10492
Kmeans based on CH index (a) ^a	2	95, 181	0.38406	0.04816
Kmeans based on CH index (b) ^a	2	103, 173	0.35542	0.07892
HDDC "BIC" (a) ^a	5	40, 119, 28, 27, 62	0.53986	0.08396
HDDC "BIC" (b) ^a	5	147, 64, 16, 46, 3	0.50362	0.06951
MixtPPCA "BIC" (a)	6	69, 55, 32, 38, 6, 76	0.63406	0.07866
MixtPPCA "BIC" (b)	3	75, 174, 27	0.46014	0.02750
PAM "Silhouette width" (a)	2	173, 103	0.36957	0.06328
PAM "Silhouette width" (b)	2	180, 96	0.37319	0.05888
FunHDDC "BIC"	1	276	0.41667	0.00000

Note: When a method is followed by letter in parentheses it denotes the following: (a) = raw-data methods, with discretized data; (b) = filtering methods using spline coefficients from 10 splines.

^aResults are based on the mean of 1000 repetitions.

The Distal data curves for the 276 images are given in Figure 5(a). It is clearly seen that there is a high level of similarity between the two classes, which makes the distinction between them difficult. Figure 5(b) displays the forward plot based on the functional spatial rank for the Distal data. Two clusters are evident with two clear peaks at 67 and the other 175. Before applying the classifier in the algorithm, 36 curves remain unassigned to a cluster and two curves have been incorporated in both clusters. After step 7, each of these 38 curves has been assigned to a single appropriate cluster and H for the algorithm is 0.236.

It is clear from Figure 5(a) that one curve of the curves (number 220), which starts from the upper left and travels in a different direction from the other curves is a potential outlier. From the data, the cut-off C for outliers equals 0.9586 and for this curve, $FSRN_{F_i}(x_{220}(t)) = 0.9726$. On this basis, curve 220 may be considered an outlier in the Distal dataset.

Table 5 gives the results for the Distal data, and as can be seen, only five methods including the FSFSR algorithm gave the correct number of clusters (2). The FSFSR algorithm again records the lowest H (23.6%). For the ARI, *mclust* method based on 10 spline coefficient has the highest ARI with 0.104 but identified four clusters.

5 Discussion

In this paper, we have proposed a new forward search algorithm for clustering functional data. It is an extension to the forward search methodology based on spatial ranks that has been introduced for the multivariate case.¹² It may be used to identify the number of clusters in the underlying functional data and does not require any preprocessing of the data, nor the need to perform data registration or dimension reduction before clustering. Furthermore, it may be used in cases when the number of variables exceeds the number of observations or when the cluster size is less than the number of variables – this contrasts traditional forward searches based on Mahalanobis distances.

An important element of the algorithm is the inclusion of a classifier. This allows the classification of all curves to an appropriate cluster, even when in the early steps some have either not been assigned or have been assigned to more than one cluster.

As the FSFSR algorithm both identifies clusters and classifies functional data, any reasonable comparison should be with methods that are also capable of clustering and classifying. Equally, it is important that the metric used to gauge performance also adequately captures both clustering and classifying. To this end, we used the misclassification rate H, which penalizes methods that identify an incorrect number of clusters as well as assessing the error in classification and the adjusted Rand index (ARI) which is a popular metric used in classification and clustering.

For the simulated examples, the algorithm was able to identify correctly the number of clusters and the number of simulated curves in each cluster with an H of no more than 0.0063. Indeed for the third, more complex simulated example, it was the only algorithm to correctly identify the number of clusters with a near perfect H and ARI score.

For the two real examples, the FSFSR algorithm identified the correct number of clusters and had the lowest H amongst all the methods. However, in the last example it also had one of the poorest ARI scores and illustrates some of the shortcomings when using these metrics for comparing algorithms. It is clear from the real data examples that when an incorrect number of clusters are returned, H penalizes algorithms more severely than the ARI. In contrast, the ARI adjusts for correct classification by chance which should, in principle, give it an advantage over H.^{26,40,41} However, since the baseline expected Rand index may be different between two different partitions of the data, it is not clear if two algorithms were to return similar values for the ARI that this would represent equivalence in performance.^{26,40,41} Thus comparing performances can be difficult using this metric.

One of the limitations in the proposed algorithm is that, in order to identify the subset size correspondence to each peak in the trajectories of the random starts, we have to find the highest $r_{min}(m)$ around each peak and set m as the cluster size. Currently, we stop the search at each peak, and then identify the subset size and its membership. However, there is the potential to automate this process using ideas contained in Cerioli et al.⁴² and this requires further research.

Several authors have demonstrated the use of the forward search based around a Mahalanobis distance metric to detect outliers on multivariate data.^{43–47} Distributional results are known for the Mahalanobis distance and the minimum Mahalanobis distance allowing inferential statements to be made. In particular, percentile envelopes that contain most of the data may be estimated so that outlier points lie outside the enveloped region. In contrast, the forward search proposed here has been developed in a nonparametric framework. This makes it more difficult to use envelopes from order statistics based on distributional assumptions and approximations for unscaled distances and asymptotic results and requires further research.

When there are a large number of clusters the proposed forward search may produce too many peaks and this may make it difficult to determine the number of clusters and their sizes. Furthermore, the selection of random starting points as used here can result in multiple peaks which makes it hard to identify small clusters; hence, a more effective divisive strategy can instead be used.^{43–47}

Although spatial ranks are invariant under orthogonal transformations, they are not invariant under general affine transformations of the data, thus the proposed algorithm is not affine invariant. An affine invariant version of the algorithm could be formulated based on affine invariant spatial ranks⁴⁸ and this could improve the results if the scales of the clusters were different for instance. However, this would make the algorithm computationally expensive and greatly increase the process time and as a result we did not use any affine invariant versions of spatial ranks here.

The treatment of outliers is important in cluster analysis as their presence may indicate the existence of clusters or populations not specified in the initial analysis. Equally they may arise due to errors in recording of some form. Potentially both can distort the process of cluster identification and data classification and, in the case of model-based approaches, bias the estimates of associated parameters. This has led some investigators to propose methods such as ‘trimming’ the data of outliers as part of the analysis.^{24,25}

The first part of the FSFSR algorithm, the forward search, identifies the number of clusters and their constituents. In some cases, some of the data may remain unclassified at this stage of the algorithm, as in the case of the Distal dataset. These unclassified data tend to have functional spatial rank norms ($FSRN_{F_i}(\cdot)$) that are larger than the classified data and further away from the spatial median. Although the FSFSR algorithm assigns these curves to an appropriate cluster, the implicit assumption is that the forward search has identified the correct number of clusters. Without further investigation, some of these curves could be unidentified outliers and indicate, potentially, the existence of other clusters.

Here we used the calculated upper whisker of the box plot distribution as the threshold for outliers and this identified one potential outlier in the Distal dataset. Although the source of the outlier is unclear, in itself it would be insufficient to conclude that it arose from another population. Other approaches to the detection and treatment of outliers have been described and this remains an active area of research.^{24,25,43,49}

In this study, both simulated and real datasets were used to compare the proposed algorithm with existing methods. One drawback when dealing with real data is that the identification of ‘true clusters’ is often not as clearly defined as in simulated datasets. Thus, errors in the reference classes, and the intrinsic dependence of the reference classification on the problem at hand, may diminish the effectiveness of this approach as a benchmarking procedure.^{50,51}

In this study, we have proposed the FSFSR algorithm and demonstrated its potential as a clustering and classifying method for functional data. A more extensive evaluation of its performance across a greater range of examples is clearly necessary. However, as a data-driven non-parametric method, the approach proposed here is

free from assumptions on the underlying distributions of the data and we believe it represents a significant development in functional data analysis.


Declaration of conflicting interests


The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: BHW was supported by funding from a Medical Research Council Clinician Scientist award (MR/N007999/1).

ORCID iDs

Mohammed Baragilly  <https://orcid.org/0000-0003-4775-4742>

Brian H Willis  <https://orcid.org/0000-0002-0821-8624>

References

- Ramsay JO and Silverman BW. *Functional data analysis*. 2nd ed. New York: Springer, 2005.
- Kokoszka P and Reimherr M. *Introduction to functional data analysis*. New York: Chapman & Hall, 2017.
- Müller HG. Functional modelling and classification of longitudinal data. *Scand J Statist* 2005; **32**: 223–240.
- Wang JL, Chiou JM and Müller HG. Functional data analysis. *Annual Review of Statistics and Its Application* 2016; **3**: 257–295.
- Jacques J and Preda C. Model-based clustering for multivariate functional data. *Computat Stat Data Analys* 2013; **71**: 92–106.
- Claeskens G, Hubert M, Slaets L, et al. Multivariate functional halfspace depth. *J Am Stat Assoc* 2014; **109**: 411–423.[10.1080/01621459.2013.856795]
- Bhattacharya M, Ehrenthal D and Shatkay H. Identifying growth-patterns in children by applying cluster analysis to electronic medical records. In: *IEEE international conference on bioinformatics and biomedicine (BIBM)*, Belfast, 2014, pp.348-351.
- Guisado-Clavero M, Roso-Llorach A, López-Jimenez T, et al. Multimorbidity patterns in the elderly: a prospective cohort study with cluster analysis. *BMC Geriatr* 2018; **18**, <https://doi.org/10.1186/s12877-018-0705-7>
- Hadi AS. Identifying multiple outliers in multivariate data. *J R Stat Soc Series B Stat Methodol* 1992; **54**: 761–771.
- Atkinson AC. Fast very robust methods for the detection of multiple outliers. *J Am Stat Assoc* 1994; **89**: 1329–1339.
- Atkinson AC, Riani M and Cerioli A. *Exploring multivariate data with the forward search*. New York, NY: Springer, 2004.
- Baragilly M and Chakraborty B. Determining the number of clusters using multivariate ranks. In: C Agostinelli, A Basu, P Filzmoser and D Mukherjee (eds) *Recent advances in robust statistics: theory and applications*. India: Springer, 2016, pp.17–33.
- Atkinson AC and Riani M. Exploratory tools for clustering multivariate data. *Computat Stat Data Analys* 2007; **52**: 272–285.
- Bellman R. *Dynamic programming*. Princeton, NJ: Princeton University Press, 1957.
- Brown BM. Statistical uses of the spatial median. *J R Stat Soc Series B Stat Methodol*.1983; **45**: 25–30.
- Chaudhuri P. On a geometric notion of multivariate data. *J Am Stat Assoc* 1996; **90**: 862–872.
- Serfling R. A depth function and a scale curve based on spatial quantiles. In: Y Dodge (ed.) *Statistical data analysis based on the L1-norm and related methods*. Birkhaeuser Verlag Basel: Springer, 2002, pp.25–38.
- Chakraborty A and Chaudhuri P. On data depth in infinite dimensional spaces. *Ann Inst Stat Math* 2014; **66**: 303–324.
- Araujo A and Giné E. *The central limit theorem for real and Banach valued random variables*. New York, NY: Wiley, 1980.
- Serfling R and Wijesuriya U. Depth-based nonparametric description of functional data, with emphasis on use of spatial depth. *Comput Stat Data Anal* 2017; **105**: 24–45.
- Cardot H, Cenac P and Zitt PA. Efficient and fast estimation of the geometric median in Hilbert spaces with an averaged stochastic gradient algorithm. *Bernoulli* 2013; **19**:18–43.
- Chaouch M and Goga C. Using complex surveys to estimate the L1-median of a functional variable: application to electricity load curves. *Int Stat Rev* 2012; **80**: 40–59.
- Sguera C, Galeano P and Lillo R. Spatial depth based classification for functional data. *Test* 2014; **23**: 725–750.
- Garcia-Escudero LA, Gordaliza A, Matran C, et al. A review of robust clustering methods. *Adv Data Analys Classification* 2010; **4**: 89–109.
- Rivera-Garcia D, Garcia-Escudero LA, Mayo-Iscar A, et al. Robust clustering for functional data based on trimming and constraints. *Adv Data Analysis Classification* 2019; **13**: 201–225.
- Meila M. Comparing clusterings—an information based distance. *J Multivar Anal* 2007; **98**: 873–895.

27. Hubert L and Arabie P. Comparing partitions. *J Classif* 1985; **2**: 193–218.
28. Fraley C and Raftery AE. Model-based clustering, discriminant analysis, and density estimation. *J Am Stat Assoc* 2002; **97**: 611–631.
29. Banfield JD and Raftery AE. Model-based Gaussian and non-Gaussian clustering. *Biometrics* 1993; **49**: 803–821.
30. Lloyd SP. Least square quantization in PCM. 1957. Bell Telephone Laboratories Paper. Published in journal much later: Lloyd SP. Least squares quantization in PCM. *IEEE Transact Informat Theory* 1982; **28**: 129–137.
31. Calinski RB and Harabasz J. A dendrite method for cluster analysis. *Commun Stat* 1974; **3**: 1–27.
32. Bouveyron C, Girard S and Schmid C. High dimensional data clustering. *Comput Stat Data Anal* 2007; **52**: 502–519.
33. Tipping ME and Bishop C. Mixtures of principal component analyzers. *Neural Comput* 1999; **11**: 443–482.
34. Reynolds A, Richards G, Iglesia B, et al. Clustering rules: A comparison of partitioning and hierarchical clustering algorithms. *J Math Model Algorithms* 1992; **5**: 475–504.
35. Kaufman L and Rousseeuw PJ. *Finding groups in data: an introduction to cluster analysis*. New York, NY: John Wiley and Sons, 1990.
36. Bouveyron C and Jacques J. Model-based clustering of time series in group specific functional subspaces. *Adv Data Anal Classif* 2011; **5**: 281–300.
37. Olszewski RT. *Generalized feature extraction for structural pattern recognition in time-series data*. PhD Thesis, Carnegie Mellon University, Pittsburgh, PA, 2001.
38. Chen Y, Keogh E, Hu B, et al. The UCR Time Series Classification Archive, www.cs.ucr.edu/~eamonn/time_series_data/ (2015).
39. Bagnall A and Davis L. Predictive modelling of bone age through classification and regression of bone shapes, <https://arxiv.org/abs/1406.4781> (2014).
40. Meilă M. Comparing clusterings by the variation of information. In: B Schölkopf and MK Warmuth (eds) *Learning Theory and Kernel Machines. Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer. 2003; vol 2777: 173–187.
41. Chacón JE. A close-up comparison of the misclassification error distance and the adjusted Rand index for external clustering evaluation. *ArXiv abs/1907*. 2019; **11505**.
42. Cerioli A, Farcomeni A and Riani M. Wild adaptive trimming for robust estimation and cluster analysis. *Scand J Statist* 2019; **46**: 235–256.
43. Riani M, Atkinson AC and Cerioli A. Finding an unknown number of multivariate outliers. *J R Stat Soc Series B Stat Methodol B* 2009; **71**: 447–466.
44. Johansen S and Nielsen B. Asymptotic theory of outlier detection algorithms for linear time series regression models. (With discussion). *Scand J Statist* 2016; **43**: 321–348.
45. Atkinson AC, Riani M and Cerioli A. Cluster detection and clustering with random start forward searches. *J Appl Stat* 2018; **45**: 777–798.
46. Cerioli A, Riani M, Atkinson AC, et al. The power of monitoring: how to make the most of a contaminated multivariate sample. (With discussion). *Stat Meth Appl* 2018; **27**: 559–587.
47. Riani M, Atkinson AC, Cerioli A, et al. Efficient robust methods via monitoring for clustering and multivariate data analysis. *Pattern Recog* 2019; **88**: 246–260.
48. Chakraborty B. On affine equivariant multivariate quantiles. *Ann Inst Stat Math* 2001; **53**: 380–403.
49. Febrero M, Galeano P and González-Manteiga W. Outlier detection in functional data by depth measures, with application to identify abnormal NO_x levels. *Environmetrics* 2008; **19**: 331–345.
50. Hennig C and Liao T. How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification. *J R Stat Soc Series B Stat Methodol C* 2013; **62**: 309–369.
51. Cerioli A, García-Escudero LA, Mayo-Isacar A, et al. Finding the number of normal groups in model-based clustering via constrained likelihoods. *J Comput Graph Stat* 2018; **27**: 404–416.