Clustering compositional data trajectories

F. Bruno, F. Greco
Dipartimento di Scienze Statistiche "P. Fortunati",
University of Bologna, Italy
francesca.bruno@unibo.it

Abstract

Our essay aims at studying suitable statistical methods for the clustering of compositional data in situations where observations are constituted by trajectories of compositional data, that is, by sequences of composition measurements along a domain. Observed trajectories are known as "functional data" and several methods have been proposed for their analysis.

In particular, methods for clustering functional data, known as Functional Cluster Analysis (FCA), have been applied by practitioners and scientists in many fields. To our knowledge, FCA techniques have not been extended to cope with the problem of clustering compositional data trajectories. In order to extend FCA techniques to the analysis of compositional data, FCA clustering techniques have to be adapted by using a suitable compositional algebra.

The present work centres on the following question: given a sample of compositional data trajectories, how can we formulate a segmentation procedure giving homogeneous classes? To address this problem we follow the steps described below.

First of all we adapt the well-known spline smoothing techniques in order to cope with the smoothing of compositional data trajectories. In fact, an observed curve can be thought of as the sum of a smooth part plus some noise due to measurement errors. Spline smoothing techniques are used to isolate the smooth part of the trajectory: clustering algorithms are then applied to these smooth curves.

The second step consists in building suitable metrics for measuring the dissimilarity between trajectories: we propose a metric that accounts for difference in both shape and level, and a metric accounting for differences in shape only.

A simulation study is performed in order to evaluate the proposed methodologies, using both hierarchical and partitional clustering algorithm. The quality of the obtained results is assessed by means of several indices.

Kew words: Compositional data trajectories, cluster analysis, functional data analysis, cluster validity indices, simulation study.

1 Introduction

Statistical data appear in compositional form in several contexts,. Compositional data are vectors of proportions describing the relative contributions of each category of possible outcomes to the whole.

Let $\boldsymbol{p} = \left(p_1, p_2, ..., p_C \right)$ be a C-dimensional vector where $p_i > 0$ for i=1,2,…,C and $\sum_i p_i = 1$. Due to the sum constraint and the bounded support, special techniques are required for compositional data analysis (Aitchison, 1982;1986).

In this paper, we deal with the trajectories of compositional data - that is, with sequences of composition measurements along a domain - by addressing the problem of clustering compositional data trajectories. For example, particle number concentrations and particle size distributions (14 classes between 0.3 to 20 μm) were monitored in Milan, using an optical particle counter (OPC GRIMM 1.108 "Dustcheck") and a portable meteorological station (Ferrero and others; 2007). Data consist in the measurement, along a vertical domain (height), of the composition of particle-size distribution. The detection of a clustering of trajectories may improve our knowledge of characteristic profiles under specific meteorological conditions.

Observed trajectories can be seen as "functional data" and several methods have been proposed for their analysis; for instance Ramsay and Silverman (2005) present several techniques for analysing such data, e.g. principal components analysis, linear modelling, canonical correlation analysis and cluster analysis.

In particular, methods for clustering functional data, known as Functional Cluster Analysis (FCA), have been applied in many fields (Ghigo and others, 2006; Ludwig and others, 1995). To our knowledge, FCA techniques have not been extended to deal with the problem of clustering compositional data trajectories. In order to extend FCA techniques to the analysis of compositional data, FCA clustering techniques have to be adapted by means of a suitable algebra for compositions.

This paper centres around the following question: how are we to formulate a segmentation procedure giving homogeneous classes starting from a set of K compositional data trajectories.

One might consider the measurements of K curves as vectors, and use a standard clustering algorithm (Hartigan, 1975); however, as Abraham and others observed (2003), there are two main drawbacks to this approach.

Firstly, measurements may have been taken at different points in the domain (misaligned data), or the domains themselves may be different, that is, the index set may not be exactly the same for the K curves. For example, when measurements are taken in a time interval, starting and end points may be different.

Secondly, in the presence of measurement errors, clustering methods applied to raw data do not benefit from the functional data structure. Because of the functional nature of observations, a suitable approach would be that of removing the noise due to measurement errors. In this way, the focus can be placed on the clustering of the smooth part of the observed curves.

Once the smooth part has been extracted from each of the K curves, the clustering algorithm can be applied by establishing a suitable metric, in order to evaluate a measure of dissimilarity between observational units.

The procedure for clustering functional data may be summarised as follows: smooth out curves in order to remove measurement errors; choose a metric with which to evaluate dissimilarity among the objects in question; apply a clustering algorithm and evaluate the quality of the resulting partition. All these steps have to be performed in this particular context by considering the compositional nature of our measurements.

The present paper is organized as follows: Section 2 examines how smoothing techniques can be applied to compositional functional data. Section 3 focuses on the construction of dissimilarity matrices between curves of compositional data. In Section 4, a simulation study is performed in order to evaluate the performances of different clustering algorithms. Section 5 concludes the article with a brief discussion.

2 Smoothing a compositional data trajectory

While in standard functional data analysis (FDA) each observational unit (curve) consists of a set of measurements (usually univariate measurements), in the present context the measurements refer to compositions, hence standard FDA methodologies have to be combined with compositional algebra in order to be applicable. In what follows, we introduce the notation required in order to define a compositional data trajectory. We think of an observed compositional data trajectory as a set of measurements taken along a domain $x \in \left[x_{min}; x_{max}\right]$, such as, for example altitude, depth or time. Moreover, we assume that T measurements have been taken along such a domain. In this paper, we adopt a dot as a subscript when considering the whole domain which the subscript refers to. Thus, we define the complete data matrix for a compositional trajectory as $\boldsymbol{D} = \left[\boldsymbol{x}_{\bullet}, \boldsymbol{p}_{\bullet\bullet}\right]$ whose generic *t*-th row is $\left[x_t, \boldsymbol{p}_{\bullet t}\right]$ (*t*=1,…,*T*) and contains the *C*-dimensional composition vector $\boldsymbol{p}_{\bullet t} = \left(p_{1t}, p_{2t}, ..., p_{Ct}\right)$ observed in correspondence with x_t.

An observed curve can be thought of as the sum of a smooth part and some noise due to measurement errors. In what follows, we propose an approach whereby spline smoothing is applied to compositional data.

When smoothing compositional trajectories, the sum-to-one constraint typical of compositional data has to be taken into account. In the context of compositional data analysis, the perturbation operator $\oplus$ (see Aitchison, 1986) enables us to obtain an error structure in the simplex ∇^{C-1} that is equivalent to the additive error model in the real space $\Re^{C}$ (Billheimer and others, 2001). More specifically, the observed value $\boldsymbol{p}_{\bullet t}$ can be thought of as the underlying true value $\boldsymbol{\pi}_{\bullet t}$ perturbed by an error $\boldsymbol{\xi}_{\bullet t}$, that is:

$$\boldsymbol{p}_{\bullet t} = \boldsymbol{\pi}_{\bullet t} \oplus \boldsymbol{\xi}_{\bullet t} \qquad \boldsymbol{p}_{\bullet t}, \boldsymbol{\pi}_{\bullet t}, \boldsymbol{\xi}_{\bullet t} \in \nabla^{C-1} \qquad t=1,\ldots,T \tag{1}$$

The estimate of $\boldsymbol{\pi}_{\bullet t}$, denoted by $\hat{\boldsymbol{p}}_{\bullet t}$, can be obtained by following the steps below:

1. First apply the additive log-ratio (*alr*) transformation (Aitchison, 1986) to the observed compositions;
2. Then smooth transformed data trajectories using standard smoothing techniques (*B*-spline, *p*-spline, cubic-spline, etc.).
3. In order to obtain smoothed compositions, the inverse *alr* transformation can be applied to smoothed transformed data. This procedure enables us to obtain smoothed trajectories which at each time *t*, comply with the sum-to-one constraint.

In what follows, each point is described in detail.

1. Firstly, we apply the additive log-ratio (*alr*) transformation to transfer observations from the $\left(C-1\right)$-dimensional simplex ∇^{C-1} to the $\left(C-1\right)$-dimensional Euclidean space:

$$\boldsymbol{z}_{\bullet t} = \left(z_{1t}, z_{2t}, ..., z_{(C-1)t}\right) = alr\left(\boldsymbol{p}_{\bullet t}\right) = \left[ln\left(\frac{p_{1t}}{p_{Ct}}\right), ln\left(\frac{p_{2t}}{p_{Ct}}\right), ..., ln\left(\frac{p_{(C-1)t}}{p_{Ct}}\right)\right] \tag{2}$$

By means of *alr* transformation, we transform the original $T \times C$ matrix $\boldsymbol{p}_{\bullet\bullet}$ in the ∇^{C-1} simplex into the $T \times \left(C-1\right)$ matrix $\boldsymbol{z}_{\bullet\bullet} = \left[\boldsymbol{z}_{1\bullet}, \boldsymbol{z}_{2\bullet}, ..., \boldsymbol{z}_{(C-1)\bullet}\right]$ in the $\left(C-1\right)$-dimensional Euclidean space.

2. Several smoothing techniques could be used to smooth the transformed data $\boldsymbol{z}_{\bullet\bullet}$. In this paper, we have adopted cubic spline smoothers. For a review of smoothing spline techniques see, for example, Hastie and others (2001). We introduce some essential background information following their approach.

Each column $\boldsymbol{z}_{i\bullet}$ (*i*=1,…,*C*-1) of the transformed data matrix is smoothed independently. The smoothing is achieved by looking among all functions $f\left(x\right)$ with the first two continuous derivatives, to find the one that minimizes the residual sum of squares:

$$RSS\left(f, \lambda\right) = \sum_{t=1}^{T}\left(z_{it} - f\left(x_t\right)\right)^2 + \lambda\int\left[f''\left(s\right)\right]^2 ds = \sum_{t=1}^{T}\varepsilon_{it}^2 + \lambda\int\left[f''\left(s\right)\right]^2 ds \tag{3}$$

where λ is a fixed smoothing parameters, the first term is a measure of proximity to the data and the second term penalizes curvature in the interpolating function.

It can be shown that the solution to the minimum problem is a natural cubic spline with knots at the unique values of the x_t, $t=1,\ldots,T$, which can be written as:

$$f(x) = \sum_{j=1}^{N} N_j(x)\beta_j \qquad (4)$$

where $N_j(x)$ are a set of third degree polynomial basis functions representing the family of natural splines. Estimators of the β_j parameters are obtained using a generalized ridge regression. We should point out that (4) is an example of a linear smoother, since estimated parameters $\hat{\beta}_j$ are obtained as linear combinations of the z_{it} values. Let $\boldsymbol{N}$ be a $T \times T$ matrix whose generic element is $N_{ij} = N_j(x_t)$ and let $\boldsymbol{\Omega}$ be a $T \times T$ matrix whose generic element is $\Omega_{jk} = \int N_j''(s) N_k''(s)ds$, thus:

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{N}^T\boldsymbol{N} + \lambda\boldsymbol{\Omega}\right)^{-1} \boldsymbol{N}^T \boldsymbol{z}_{i\bullet} \qquad (5)$$

where $\hat{\boldsymbol{\beta}} = \left(\hat{\beta}_1, \hat{\beta}_2, ..., \hat{\beta}_T\right)$. The vector of fitted (smoothed) values $\hat{z}_{i\bullet}$ obtained in correspondence to the predictor vector $\boldsymbol{x}_{\bullet}$ can be expressed as:

$$\hat{\boldsymbol{z}}_{i\bullet} = \boldsymbol{N}\hat{\boldsymbol{\beta}} = \boldsymbol{N}\left(\boldsymbol{N}^T\boldsymbol{N} + \lambda\boldsymbol{\Omega}\right)^{-1} \boldsymbol{N}^T \boldsymbol{z}_{i\bullet} = \boldsymbol{S}_{\lambda}\boldsymbol{z}_{i\bullet} \qquad (6)$$

The linear operator $\boldsymbol{S}_{\lambda}$ is known as the smoother matrix. The effective degrees of freedom df_{λ} of the smoothing spline are captured by the trace of this matrix. A smoothing spline can be equivalently parameterised in terms of degrees of freedom or in terms of the λ smoothing parameter, since $df_{\lambda} = tr(\boldsymbol{S}_{\lambda})$ is a monotone function of λ. Both parameters govern the amount of smoothing in the estimated function. Once the functions $N_j(x)$ have been chosen, the smoothing spline is fully defined by the degrees of freedom (or equivalently by the smoothing parameter). Then the problem of selecting a spline function come down to selecting the number of degrees of freedom. As an example, different df_{λ} values might be tested, and the selection might be based on approximate *F*-tests, residual plots and other more subjective criteria.

By means of (6), transformed observed values z_{it} at each observed value of the predictor x_t, can be expressed as the sum of a smooth part and a noise as follows:

$$z_{it} = \hat{f}(x_t) + \varepsilon_{it} = \hat{z}_{it} + \varepsilon_{it} \qquad (7)$$

Once the smoothing spline has been estimated, prediction at every point in the interval $[x_{min}; x_{max}]$ can be obtained at each desired value of the predictor variable. This turns out to be very useful when comparing different trajectories measured at different *x* values in the interval $[x_{min}; x_{max}]$. Let $\tilde{x}$ be the new point where a prediction is needed. Prediction is obtained as:

$$\tilde{z}_i = \boldsymbol{N}(\tilde{x})^T \hat{\boldsymbol{\beta}} \qquad (8)$$

where $\boldsymbol{N}(\tilde{x})$ is the $T \times 1$ vector of the basis function evaluated at $\tilde{x}$.
3. Finally, we obtain the smoothed composition trajectory by means of the inverse *alr* transformation:

$$\hat{\boldsymbol{p}}_{\bullet t} = alr^{-1}(\hat{\boldsymbol{z}}_{\bullet t}) = \left(\frac{exp(\hat{z}_{1t})}{1+\sum_{i=1}^{C-1}\hat{z}_{it}}, ..., \frac{exp\left(\hat{z}_{(C-1)t}\right)}{1+\sum_{i=1}^{C-1}\hat{z}_{it}}, \frac{1}{1+\sum_{i=1}^{C-1}\hat{z}_{it}} \right) \qquad (9)$$

An example of a smoothed trajectory when *C*=3 is shown in Figure 1.

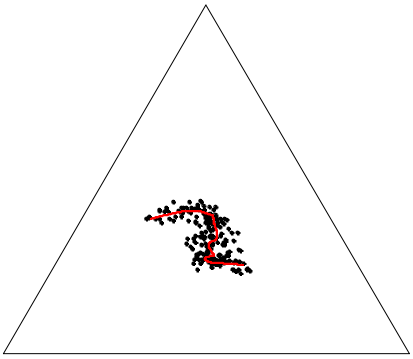

Figure 1: An example of a smoothed compositional trajectory.

3 Functional Cluster Analysis for compositional data

The clustering problem consists in partitioning a given data set into groups (clusters) so that the data points in a cluster are more similar to each other than points in different clusters (Guha and others, 1998). Thus the main concern in the clustering process is to portray the organization of patterns into "sensible" groups, in such a way that it reveals similarities and differences, and how to derive useful conclusions from them. This section proposes one method of clustering *K* compositional trajectories.

Let p_{itk} be the *i*-th class proportion observed in correspondence of the *t*-th measurement for the *k*-th trajectory, where *i*=1,..,*C*, *t*=1,…,T_k and *k*=1,…,*K*. Here T_k denotes the number of observations for the *k*-th trajectory and the subscript *k* is used in order to emphasise the fact that different numbers of observations may be available for each trajectory.

In keeping with the notation introduced in section 2, we define the data matrix related to the *k*-th trajectory as $\boldsymbol{D}_k = \left[\boldsymbol{x}_{\bullet k}, \boldsymbol{p}_{\bullet\bullet k}\right]$. Here $\boldsymbol{p}_{\bullet tk}$ denotes the *C*-dimensional compositional vector at time *t* for trajectory *k*, while $\boldsymbol{p}_{\bullet\bullet k}$ denotes the $T_k \times C$ matrix containing data measured in the *k*-th trajectory. We obtain smoothed compositional trajectories $\hat{\boldsymbol{p}}_{\bullet\bullet k}$, for *k*=1,..,*K*, following the steps proposed in section 2.

In order to cluster observed trajectories, we need to evaluate differences in the smoothed trajectories $\hat{\boldsymbol{p}}_{\bullet\bullet k}$. Several applications in functional cluster analysis are based on the measurement of dissimilarities in observed curves, by evaluating differences between the spline coefficients $\hat{\boldsymbol{\beta}}_k = \left(\hat{\beta}_{1k}, \hat{\beta}_{2k}, ..., \hat{\beta}_{Tk}\right)$ (Abraham and others, 2003; Ghigo and others, 2006). This approach is appropriate only if the same basis functions degree and knots placement are used for each curve. Since measurements could be taken for different values of the predictor variable (misalignment) and the quantities $min\left(\boldsymbol{x}_{\bullet k}\right)$ and $max\left(\boldsymbol{x}_{\bullet k}\right)$ may vary substantially among trajectories, we prefer a more flexible approach based on different knots placement for each trajectory. As a consequence, smoothed trajectories cannot be compared in terms of $\hat{\boldsymbol{\beta}}_k$ coefficients.

3.1 Building a suitable metric

In order to compare the *K* trajectories, a measure of the distances between them has to be created. A proximity measure and a clustering criterion are the main features of a clustering algorithm. The proximity measure quantifies the "similarity" between two data points.

Given two generic functions *f* and *g*, a measure of the distance between them in the interval $\left[x_{min}; x_{max}\right]$ is the integral:

$$d\left(f,g\right) = \int_{X_{max}}^{X_{min}} \left\| f\left(x\right) - g\left(x\right) \right\| dx \qquad (10)$$

where $\|\bullet\|$ denotes a norm. In what follows, we propose a strategy for evaluating this integral when the two functions lie in the simplex ∇^{C-1}.

First of all, we propose to approximate integral (10) using Monte Carlo integration by averaging point-to-point distances on a regular grid in the interval $\left[x_{min};x_{max}\right]$ as follows:

$$d\left(f,g\right)\cong n^{-1}\sum\nolimits_{i=1}^{n}\left\|f\left(x_i\right)-g\left(x_i\right)\right\| \tag{11}$$

This Monte Carlo integration can be made arbitrarily accurate by increasing the number *n* of points.
This approximation is used in order to evaluate the distance between the trajectories *l* and *k*, *l*,*k*=1,…,*K*. Starting from the observed values $\boldsymbol{p}_{\bullet\bullet l}$ and $\boldsymbol{p}_{\bullet\bullet k}$, we obtain smoothed trajectories $\hat{\boldsymbol{p}}_{\bullet\bullet l}$ and $\hat{\boldsymbol{p}}_{\bullet\bullet k}$ by means of the procedure outlined in Section 2. Approximation (11) to integral (10) can be evaluated when data are realigned on a regular grid (x_i; $i=1,...,n$). This can be done by obtaining predicted values $\tilde{\boldsymbol{p}}_{\bullet\bullet l}$ and $\tilde{\boldsymbol{p}}_{\bullet\bullet k}$ at each x_i point as described in Equation (8). Unlike $\boldsymbol{p}_{\bullet\bullet l}$ and $\boldsymbol{p}_{\bullet\bullet k}$, these predicted values are then aligned on the grid x_i; $i=1,...,n$, and thus (11) can be evaluated once a suitable norm in the simplex has been chosen.

Given two compositions $\boldsymbol{q}=\left(q_1,q_2,...,q_C\right)$ and $\boldsymbol{w}=\left(w_1,w_2,...,w_C\right)$ the difference between such vectors is evaluated as:

$$\boldsymbol{q}\Theta\boldsymbol{w}=\Gamma\left[\frac{q_1}{w_1},\frac{q_2}{w_2},...,\frac{q_C}{w_C}\right]=\left[\frac{q_1/w_1}{\sum_{i=1}^{C}q_i/w_i},\frac{q_2/w_2}{\sum_{i=1}^{C}q_i/w_i},...,\frac{q_C/w_C}{\sum_{i=1}^{C}q_i/w_i}\right]=\boldsymbol{m} \tag{12}$$

where Γ is the constraining operator which transforms each vector with positive components into a unit-sum vector. Therefore we define the distance $d\left(\boldsymbol{q},\boldsymbol{w}\right)$ as the norm of the difference (12), as defined in Billheimer and others (2001):

$$d\left(\boldsymbol{q},\boldsymbol{w}\right)=\left\|\boldsymbol{m}\right\|=alr\left(\boldsymbol{m}\right)'\boldsymbol{\Psi}^{-1}alr\left(\boldsymbol{m}\right) \tag{13}$$

where $\boldsymbol{\Psi}^{-1}=\left[\boldsymbol{I}_{C-1}-\frac{1}{C}\left(\boldsymbol{j}_{C-1}\boldsymbol{j}'_{C-1}\right)\right]$, $\boldsymbol{I}_{C-1}$ is a (*C*-1)-dimensional identity matrix, and $\boldsymbol{j}_{C-1}$ is a (*C*-1)-dimensional column vector of ones. The norm (13) meets the requirements of scale invariance and permutation invariance discussed in Aitchison (1992).
By using Equation (13) we estimate the distance between trajectories *k* and *l* as:

$$d\left(l,k\right)\cong n^{-1}\sum\nolimits_{i=1}^{n}\left\|\tilde{\boldsymbol{p}}_{\bullet il}-\tilde{\boldsymbol{p}}_{\bullet ik}\right\| \tag{14}$$

By calculating (14) for each pair *l*,*k* with *l*,*k*=1,…,*K*, we obtain the distance matrix $\boldsymbol{D}$ whose generic entry is $\boldsymbol{D}_{lk}=d\left(l,k\right)$. Starting from this matrix, alternative clustering algorithms can be adopted.

When metric (13) is used, differences between observed curves will depend on differences in their shapes and on the distance between their centers in the simplex. In an Euclidean space, this is the same as saying that the value of integral (10) depends on the norm of the difference between means $\overline{f}$ and $\overline{g}$, as well as on the integral of the difference of the centered functions $f\left(x\right)-\overline{f}$ and $g\left(x\right)-\overline{g}$, that is:

$$d\left(f,g\right)=\left(x_{max}-x_{min}\right)\left\|\overline{f}-\overline{g}\right\|+\int_{x_{min}}^{x_{max}}\left\|\left[f\left(x\right)-\overline{f}\right]-\left[g\left(x\right)-\overline{g}\right]\right\|dx \tag{15}$$

If the aim is to measure differences in shapes, only the second term of Equation (15) should be considered. The same argument applies when dealing with curves in the simplex. The center of a curve in the simplex is represented by its geometric mean. Thus, given predicted values $\tilde{\boldsymbol{p}}_{\bullet\bullet k}$, k=1,…,K, we obtain centered trajectories as:

$$\tilde{\boldsymbol{c}}_{\bullet\bullet k} = \tilde{\boldsymbol{p}}_{\bullet\bullet k} \Theta \tilde{\boldsymbol{g}}_k \tag{16}$$

where $\tilde{\boldsymbol{g}}_k = \left(\prod_{i=1}^{n} \tilde{\boldsymbol{p}}_{\bullet ik}\right)^{n^{-1}}$ is the geometric mean of the predicted values for trajectory k.
Then we propose the distances between centered trajectories as a suitable metric:

$$d^*\left(l,k\right) \cong n^{-1} \sum_{i=1}^{n} \left\| \tilde{\boldsymbol{c}}_{\bullet il} - \tilde{\boldsymbol{c}}_{\bullet ik} \right\| \tag{17}$$

By calculating (17) for each pair l,k with l,k=1,…,K, we obtain the distance matrix $\boldsymbol{D}^*$ whose generic entry is $\boldsymbol{D}_{lk}^* = d^*\left(l,k\right)$.

3.2 Cluster analysis and cluster validity assessment

A multitude of clustering methods have been proposed in the literature. The algorithms can be broadly classified into the following types (Jain and others, 1999): partitional clustering, hierarchical clustering, density-based clustering and grid-based clustering. This paper considers the first two types of algorithm.

One example of a partitional algorithm is PAM (Partitioning Around Medoids). An appealing use of the PAM algorithm is to determine a representative object (medoid) for each cluster, that is, to find the most centrally located objects within clusters. The algorithm begins by selecting an object as a medoid for each cluster. Then each of the non-selected objects is grouped together with the medoid it most closely resembles. PAM swaps medoids with other non-selected objects until all objects qualify as medoids.

Hierarchical clustering algorithms can further be divided into (Theodoridis and Koutroubas, 1999) agglomerative and divisive algorithms, depending on the method used to produce the clusters. An example of agglomerative hierarchical clustering is the Ward clustering algorithm (Ward, 1963), which works bottom up, merging objects at each step in order to obtain the minimum within-group increase in variance.

The objective of clustering methods is to discover the existence of significant groups within a given data set. In general, they search for clusters whose members are close to each other (in other words, that display considerable similarity) and clearly separated.

One of the most important issues in cluster analysis is the evaluation of clustering results to find the type of partitioning that best fits the underlying data. This is the main subject of cluster validity. We are now going to discuss some fundamental concepts within this topic, and examine certain cluster validity approaches proposed in the literature.

We define the term “optimal” clustering scheme as being the outcome of running a clustering algorithm that best fits the inherent partitions of the data set. The problems of deciding which number of clusters best fits the data set, and of evaluating clustering results, have been the subject of several studies.

The procedure of evaluating the results of a clustering algorithm is known as “cluster validity” assessment. Our cluster validity assessment is based on *external criteria* as described in Theodoridis and Koutroubas (1999). According to these criteria, clustering results are evaluated on the basis of a pre-specified structure, and this make such criteria particularly well-suited for simulation studies where the membership of each object to a cluster is specified beforehand in the simulation study design.

A number of validity indices have been defined and proposed in literature (Halkidi and others, 2000 and 2001). The indexes are built by comparing the proximity matrix obtained using a clustering algorithm $\tilde{\boldsymbol{P}}$, with the true proximity matrix $\boldsymbol{P}$ computed on the basis of the designed simulation study. When clustering K objects, the proximity matrix is a symmetric $K \times K$ dimensional matrix whose generic element $P_{ij} = 1$ if objects i and j are in the same cluster, or $P_{ij} = 0$ otherwise.

For building appropriate indices, the following quantities have to be defined. Let:

- *a* be the number of pairs for which $\tilde{P}_{ij} = 1$ and $P_{ij} = 1$.
- *b* be the number of pairs for which $\tilde{P}_{ij} = 1$ and $P_{ij} = 0$.
- *c* be the number of pairs for which $\tilde{P}_{ij} = 0$ and $P_{ij} = 1$.
- *d* be the number of pairs for which $\tilde{P}_{ij} = 0$ and $P_{ij} = 0$.

We adopt the following indices in order to evaluate cluster validity:

- *Rand Statistic:* $R = \frac{2(a+d)}{K(K-1)}$
- *Jaccard Coefficient:* $J = a/(a+b+c)$
- *Folkes and Mallows index:* $FM = \sqrt{\frac{a}{a+b} \cdot \frac{a}{a+c}}$

For the previous three indices, it has been proven that high values of indices indicate great similarity between $\tilde{\boldsymbol{P}}$ and $\boldsymbol{P}$. The higher the values of these indices, the higher the quality of the results.
See Halkidi and others (2001) for a general examination of the properties of such indices.

4 Simulation study

In order to evaluate the performances of the proposed methods, we have performed a simulation study and compared the results obtained with PAM and Ward clustering algorithms applied by utilising $\boldsymbol{D}$ and $\boldsymbol{D}^*$ distance matrices. For the sake of simplicity, let *C*=3.

The simulation study is designed to evaluate the effect of the shape-difference and level-difference in the clustering procedures. In particular, we are interested in comparing results obtained using $\boldsymbol{D}$ and $\boldsymbol{D}^*$ dissimilarity matrices. For this reason, several settings have been designed as follows. The trajectories are determined as a third grade polynomial on the *alr* scale in the interval $x \in [x_{min}; x_{max}]$.

4.1 Population specification

For each setting we generate observed trajectories from four populations: each population is created as the combination of different centers and shapes.
The population shapes are determined as third degree polynomial functions on the *alr* scale with coefficients matrix $\boldsymbol{\delta} = \{\delta_{ij};\ i = 1,2,3;\ j = 1,2\}$:

$$\begin{aligned} z_{1t} &= \delta_{11}x_t + \delta_{12}x_t^2 + \delta_{13}x_t^3 \\ z_{2t} &= \delta_{21}x_t + \delta_{22}x_t^2 + \delta_{23}x_t^3 \end{aligned} \tag{18}$$

and then back-transformed using $alr^{-1}(\boldsymbol{z}_{\bullet t})$. In particular, the specified values for each setting are reported in Table 1, where $\boldsymbol{\delta}$ coefficients are indexed by a superscript denoting the population shape (A and B) to which the coefficients refer.

In order to take the differences in centers into consideration, we consider three pairs of different compositional centers (Table 2).

By perturbing shapes (Table 1) with centers (Table 2), we generate nine different settings: each setting is characterised by the different importance of shape-difference and level-difference on population dissimilarity. True population values are denoted as $\boldsymbol{\pi}_{\bullet t}^{L}$, where *t*=1,…,*T* and the label *L* indicates the populations.

$$\begin{aligned} \boldsymbol{\pi}_{\bullet t}^{A.1} &= alr^{-1}\left[\boldsymbol{z}_{\bullet t}\left(\boldsymbol{\delta}^{A}\right)\right] \oplus \mu_1 \qquad & \boldsymbol{\pi}_{\bullet t}^{A.2} &= alr^{-1}\left[\boldsymbol{z}_{\bullet t}\left(\boldsymbol{\delta}^{A}\right)\right] \oplus \mu_2 \\ \boldsymbol{\pi}_{\bullet t}^{B.1} &= alr^{-1}\left[\boldsymbol{z}_{\bullet t}\left(\boldsymbol{\delta}^{B}\right)\right] \oplus \mu_1 \qquad & \boldsymbol{\pi}_{\bullet t}^{B.2} &= alr^{-1}\left[\boldsymbol{z}_{\bullet t}\left(\boldsymbol{\delta}^{B}\right)\right] \oplus \mu_2 \end{aligned} \tag{19}$$

Table 1. Shape parameters.

Setting 1	Shape 1	$\delta_{11}^A,\delta_{12}^A,\delta_{13}^A$	0.6	0.3	-0.36
		$\delta_{21}^A,\delta_{22}^A,\delta_{23}^A$	0.33	0.6	0.04
	Shape 2	$\delta_{11}^B,\delta_{12}^B,\delta_{13}^B$	1	0.5	-1
		$\delta_{21}^B,\delta_{22}^B,\delta_{23}^B$	0.53	1	0.11
Setting 2	Shape 1	$\delta_{11}^A,\delta_{12}^A,\delta_{13}^A$	0.4	0.4	-0.2
		$\delta_{21}^A,\delta_{22}^A,\delta_{23}^A$	-0.6	0.5	0
	Shape 2	$\delta_{11}^B,\delta_{12}^B,\delta_{13}^B$	1	0.5	-1
		$\delta_{21}^B,\delta_{22}^B,\delta_{23}^B$	0.53	1	0.11
Setting 3	Shape 1	$\delta_{11}^A,\delta_{12}^A,\delta_{13}^A$	0.9	0.2	-0.8
		$\delta_{21}^A,\delta_{22}^A,\delta_{23}^A$	0.55	-0.2	0.2
	Shape 2	$\delta_{11}^B,\delta_{12}^B,\delta_{13}^B$	1	0.5	-1
		$\delta_{21}^B,\delta_{22}^B,\delta_{23}^B$	0.53	1	0.11

Table 2. Center parameters.

Setting X.1	μ_1	0.7	0.2	0.1
	μ_2	0.5	0.3	0.2
Setting X.2	μ_1	0.7	0.2	0.1
	μ_2	0.65	0.25	0.1
Setting X.3	μ_1	0.7	0.2	0.1
	μ_2	0.69	0.19	0.12

For example, the first population for Setting 1.1 has center μ_1 referring to row Setting X.1 in Table 2, and Shape 1 referring to row Setting 1 in Table 1. The second population for Setting 1.1 has center μ_2 referring to row Setting X.1 in Table 2 and Shape 2 referring to row Setting 1 in Table 1. The third population for Setting 1.1 has center μ_2 referring to row Setting X.1 in Table 2 and Shape 1 referring to row Setting 1 in Table 1. The fourth population for Setting 1.1 has center μ_1 referring to row Setting X.1 in Table 2 and Shape 2 referring to row Setting 1 in Table 1. Population trajectories for each setting are illustrated in Figure 2.

We would point out that three different proximity matrix may be created:

- $\boldsymbol{P}^{Shape}$: whose generic element $P_{ij}^{Shape}=1$ if population trajectories *i* and *j* have the same shape,
- $\boldsymbol{P}^{Center}$: whose generic element $P_{ij}^{Center}=1$ if population trajectories i and j have the same center,
- $\boldsymbol{P}$: whose generic element $P_{ij}=1$ if population trajectories i and j have the same center and shape.

Matrices $\boldsymbol{P}^{Shape}$ and $\boldsymbol{P}^{Center}$ describe the proximity structure of a population with two clusters only, whereas $\boldsymbol{P}$ refers to a population with four clusters.

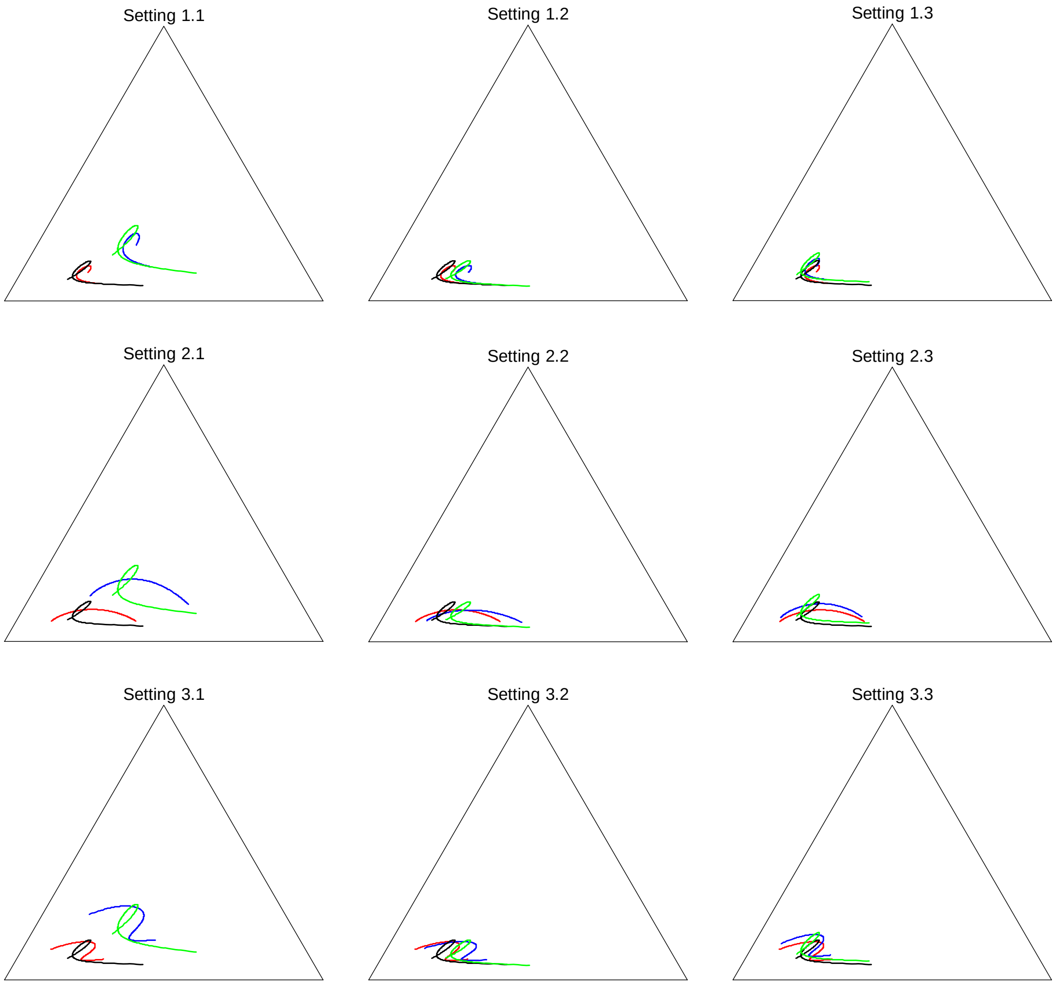

Figure 2: Population trajectories for all settings.

4.2 Simulating data sets

We consider a population trajectory to be the law underlying observed values. Given a true population to which a trajectory belongs, we assume that an observed trajectory is characterised by three source of variability, which cause discrepancies between observed data and the true underlying law. The first two sources of variability are tied to the variations that are going to characterise centers and shapes due to the natural variability of the phenomenon. The third sourced of variability is associated with measurement error. All these sources of variability are reproduced throughout the simulation study.

A number of data sets are simulated. Each data set comprises *K*=100 trajectories: the first 20 are generated from the first population in the setting. For the other three populations in question, we generate 15, 40 and 25 trajectories respectively. We are now going to describe how a generic trajectory is generated, ignoring the indexes referring to the population to which the trajectory refers, in order to simplify notation.

Each trajectory is generated by adding noise to coefficients and centers in Tables 1-2.

Let $\boldsymbol{\mu}_{\bullet k}$ be the center of the *k*-th simulated trajectory. In order to obtain this center for a simulated trajectory, we perturbe population center $\boldsymbol{\mu}_{\bullet} = (\mu_1, \mu_2, \mu_3)$, by adding noise $\boldsymbol{\varepsilon}_{\mu}$ on the *alr* scale as follows:

$$alr(\boldsymbol{\mu}_{\bullet k}) = alr(\boldsymbol{\mu}_{\bullet}) + \boldsymbol{\varepsilon}_{\mu} \qquad \varepsilon_{\mu_i} \sim N(0, \sigma^2_{\mu_i}) \qquad i=1,2 \tag{20}$$

The variance of the errors $\sigma^2_{\mu_i}$ is fixed in order to control the coefficient of variation of the centers ${}_k\boldsymbol{\mu}$, and is held constant throughout the simulation study.
Then the coefficients determining the shape of the *k*-th trajectory are obtained by perturbing coefficients in Equation (18) using $\varepsilon_{\delta_{ij}}$ error:

$$\delta_{ijk} = \delta_{ij} + \varepsilon_{\delta_{ij}} \qquad \varepsilon_{\delta_{ij}} \sim N\left(0,\sigma^2_{\delta_{ij}}\right) \quad i=1,2;\ j=1,2,3;\ k=1,\ldots,100 \tag{21}$$

The variance in the errors $\sigma^2_{\delta_{ij}}$ is fixed in order to control the coefficient of variation (*CV*) of the ${}_k\delta_{ij}$ coefficients. The nine settings are studied using four different values of such *CV*s (respectively .10, .15, .20, .25). A total of *M*=200 data sets are simulated for each setting, 50 for each *CV* value.
Finally measurement error is added by means of the terms $\boldsymbol{\upsilon}_{\bullet t}$ in order to obtain simulated values ${}_k\boldsymbol{p}_{\bullet t}$, *t*=1,…,*T*.

$$\boldsymbol{p}_{\bullet tk} = alr^{-1}\left(\boldsymbol{z}_{\bullet t}\left(\boldsymbol{\delta}_k\right)\right) \oplus \boldsymbol{\mu}_k \oplus \boldsymbol{\upsilon}_{\bullet t} \tag{22}$$

The first two terms in Equation (22) are the population values indicated in Equation (19) perturbed by means of Equations (20) and (21).

4.3 Results

In Table 3(a)-3(d) simulation results are reported separately for each value of the *CV*s (respectively 0.1, 0.15, 0.2 and 0.25) of the δ_{ijk} coefficients in Equation (21). Each Table comprise four sub-tables indicated by roman numerals, and the Rand (*R*), Jaccard (*J*) and Falks and Mallows (*FM*) indices are reported for each sub-table. The following is a description of each sub-table:
(I): Results obtained by using the $\boldsymbol{D}^*$ dissimilarity matrix, choosing two clusters and comparing the resulting $\tilde{\boldsymbol{P}}^{Shape}$ proximity matrix with the $\boldsymbol{P}^{Shape}$ population proximity matrix. This enables us to obtain a clustering structure when only shape dissimilarity is of interest.
(II): Results obtained by using the $\boldsymbol{D}$ dissimilarity matrix, choosing two clusters and comparing the resulting $\tilde{\boldsymbol{P}}^{Shape}$ proximity matrix with the $\boldsymbol{P}^{Shape}$ population proximity matrix.
(III): Results obtained by using the $\boldsymbol{D}$ dissimilarity matrix, choosing two clusters and comparing the resulting $\tilde{\boldsymbol{P}}^{Center}$ proximity matrix with the $\boldsymbol{P}^{Center}$ population proximity matrix.
Sub-tables (II) and (III) are designed to establish which clustering structure (the one discriminated more by shape or the one discriminated more by the center) is captured when two clusters are sought.
(IV): Results obtained by using the $\boldsymbol{D}$ dissimilarity matrix, choosing four clusters and comparing the resulting $\tilde{\boldsymbol{P}}$ proximity matrix with the $\boldsymbol{P}$ population proximity matrix.

Tables 4(a)-4(d) show the equivalent results obtained using the Ward Algorithm.
First of all, according to the values of *R*, *J* and *FM* indices, no appreciable differences are found between the quality of results obtained using PAM (Table 3) and Ward (Table 4) algorithms. For this reason, we have chosen to comment on the results in Table 3 only.

In sub-tables (I)a-d, for all CV values, the considered indexes perform well in representing the clustering structure according to shape: i.e. $\tilde{\boldsymbol{P}}^{Shape}$ and $\boldsymbol{P}^{Shape}$ are very similar. As CV values increase, i.e. more noise on the shape parameters is introduced into the simulation, the values of the quality indices fall. In particular, for Setting 1 (characterised by similar shapes) and CV=0.25, we observe the lowest values of quality indices.
In sub-tables (IV), when 4 clusters are sought, low-quality clustering structure is obtained because trajectories are not sufficiently discriminated in terms of shape or center. We observe a particular performance of indices in setting 2.1, characterised by substantial differences in both shape and level.
As regards sub-tables (II) and (III), we note that for settings 1.1, 2.1 and 3.1, where the center-difference is predominant, if the $\boldsymbol{D}$ metric is adopted, the two clusters obtained reproduce the proximity structure $\boldsymbol{P}^{Center}$. As regards Settings 1.2, 1.3, 3.2 and 3.3, low quality clustering structure is obtained in both sub-tables (II) and (III). This shows that, when 4 clusters cannot be detected, the $\boldsymbol{D}^*$ metric proves far more effective in capturing clusters of shapes, since it is cleansed of the confounding effect of difference in centers.

Table 3. Simulation results. PAM algorithm.

(a)	**(I)**			**(II)**			**(III)**			**(IV)**		
	R	***J***	***FM***	***R***	***J***	***FM***	***R***	***J***	***FM***	***R***	***J***	***FM***
Setting1.1	1.00	1.00	1.00	0.50	0.36	0.53	1.00	0.99	1.00	0.76	0.39	0.56
Setting1.2	1.00	1.00	1.00	0.51	0.37	0.54	0.61	0.46	0.62	0.68	0.26	0.41
Setting1.3	1.00	1.00	1.00	0.51	0.37	0.54	0.54	0.39	0.55	0.67	0.24	0.39
Setting2.1	1.00	1.00	1.00	0.50	0.36	0.53	1.00	1.00	1.00	1.00	1.00	1.00
Setting2.2	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.36	0.53	0.78	0.43	0.60
Setting2.3	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.36	0.53	0.76	0.40	0.57
Setting3.1	1.00	1.00	1.00	0.50	0.36	0.53	1.00	0.99	1.00	0.87	0.65	0.77
Setting3.2	1.00	1.00	1.00	0.61	0.48	0.64	0.58	0.43	0.60	0.76	0.39	0.55
Setting3.3	1.00	1.00	1.00	0.81	0.73	0.82	0.50	0.36	0.53	0.74	0.35	0.52

(b)	**(I)**			**(II)**			**(III)**			**(IV)**		
	R	***J***	***FM***	***R***	***J***	***FM***	***R***	***J***	***FM***	***R***	***J***	***FM***
Setting1.1	0.98	0.97	0.98	0.50	0.36	0.53	1.00	0.99	1.00	0.75	0.38	0.55
Setting1.2	0.99	0.98	0.99	0.50	0.36	0.53	0.62	0.46	0.63	0.68	0.26	0.41
Setting1.3	0.99	0.99	0.99	0.51	0.37	0.54	0.54	0.38	0.55	0.67	0.24	0.38
Setting2.1	1.00	1.00	1.00	0.50	0.36	0.53	1.00	0.99	1.00	1.00	0.99	1.00
Setting2.2	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.36	0.53	0.78	0.44	0.61
Setting2.3	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.36	0.53	0.77	0.40	0.57
Setting3.1	1.00	1.00	1.00	0.50	0.36	0.53	0.99	0.99	0.99	0.86	0.62	0.75
Setting3.2	1.00	1.00	1.00	0.60	0.47	0.63	0.59	0.44	0.61	0.75	0.37	0.54
Setting3.3	1.00	1.00	1.00	0.83	0.74	0.84	0.50	0.36	0.53	0.74	0.35	0.52

(c)	**(I)**			**(II)**			**(III)**			**(IV)**		
	R	***J***	***FM***	***R***	***J***	***FM***	***R***	***J***	***FM***	***R***	***J***	***FM***
Setting1.1	0.93	0.88	0.93	0.50	0.36	0.53	1.00	0.99	1.00	0.76	0.40	0.57
Setting1.2	0.93	0.88	0.93	0.51	0.36	0.53	0.62	0.46	0.63	0.68	0.25	0.39
Setting1.3	0.91	0.85	0.91	0.52	0.38	0.55	0.53	0.38	0.55	0.67	0.24	0.38
Setting2.1	1.00	1.00	1.00	0.50	0.36	0.53	0.99	0.99	0.99	1.00	0.99	1.00
Setting2.2	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.36	0.53	0.78	0.43	0.60
Setting2.3	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.36	0.53	0.77	0.40	0.57
Setting3.1	1.00	0.99	1.00	0.50	0.36	0.53	1.00	0.99	1.00	0.84	0.56	0.71
Setting3.2	0.99	0.99	0.99	0.61	0.49	0.64	0.58	0.43	0.60	0.75	0.36	0.53
Setting3.3	1.00	0.99	1.00	0.73	0.62	0.74	0.52	0.37	0.54	0.73	0.34	0.51

(d)	**(I)**			**(II)**			**(III)**			**(IV)**		
	R	***J***	***FM***	***R***	***J***	***FM***	***R***	***J***	***FM***	***R***	***J***	***FM***
Setting1.1	0.78	0.68	0.79	0.50	0.36	0.53	1.00	0.99	1.00	0.75	0.38	0.55
Setting1.2	0.73	0.60	0.74	0.51	0.36	0.53	0.62	0.46	0.63	0.66	0.23	0.37
Setting1.3	0.77	0.66	0.78	0.51	0.37	0.54	0.54	0.39	0.56	0.65	0.22	0.36
Setting2.1	1.00	1.00	1.00	0.50	0.36	0.53	1.00	0.99	1.00	1.00	0.99	1.00
Setting2.2	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.36	0.53	0.78	0.44	0.60
Setting2.3	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.36	0.53	0.76	0.40	0.57
Setting3.1	0.98	0.96	0.98	0.50	0.36	0.53	1.00	0.99	1.00	0.85	0.57	0.72
Setting3.2	0.97	0.95	0.98	0.61	0.48	0.64	0.59	0.44	0.61	0.73	0.34	0.50
Setting3.3	0.98	0.96	0.98	0.70	0.58	0.72	0.51	0.36	0.53	0.73	0.33	0.50

Table 4. Simulation results. Ward algorithm.

(a)	**(I)**			**(II)**			**(III)**			**(IV)**		
	R	***J***	***FM***	***R***	***J***	***FM***	***R***	***J***	***FM***	***R***	***J***	***FM***
Setting1.1	1.00	1.00	1.00	0.50	0.36	0.53	1.00	0.99	1.00	0.75	0.39	0.56
Setting1.2	1.00	1.00	1.00	0.51	0.38	0.55	0.60	0.45	0.62	0.66	0.24	0.38
Setting1.3	1.00	1.00	1.00	0.51	0.38	0.55	0.53	0.39	0.56	0.64	0.22	0.36
Setting2.1	1.00	1.00	1.00	0.50	0.36	0.53	1.00	0.99	1.00	1.00	0.99	1.00
Setting2.2	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.36	0.53	0.78	0.45	0.62
Setting2.3	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.36	0.53	0.76	0.40	0.57
Setting3.1	1.00	1.00	1.00	0.50	0.36	0.53	1.00	1.00	1.00	0.86	0.61	0.75
Setting3.2	1.00	1.00	1.00	0.57	0.44	0.60	0.57	0.42	0.59	0.76	0.40	0.57
Setting3.3	1.00	1.00	1.00	0.67	0.56	0.69	0.53	0.38	0.55	0.74	0.36	0.53

(b)	**(I)**			**(II)**			**(III)**			**(IV)**		
	R	***J***	***FM***	***R***	***J***	***FM***	***R***	***J***	***FM***	***R***	***J***	***FM***
Setting1.1	1.00	1.00	1.00	0.50	0.36	0.53	1.00	0.99	1.00	0.75	0.38	0.55
Setting1.2	1.00	1.00	1.00	0.51	0.38	0.55	0.59	0.45	0.62	0.65	0.23	0.38
Setting1.3	1.00	1.00	1.00	0.52	0.38	0.55	0.54	0.39	0.56	0.65	0.23	0.37
Setting2.1	1.00	1.00	1.00	0.50	0.36	0.53	1.00	0.99	1.00	1.00	0.99	1.00
Setting2.2	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.36	0.53	0.78	0.46	0.63
Setting2.3	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.36	0.53	0.76	0.40	0.57
Setting3.1	1.00	1.00	1.00	0.50	0.36	0.53	1.00	1.00	1.00	0.85	0.60	0.74
Setting3.2	1.00	1.00	1.00	0.54	0.41	0.58	0.61	0.46	0.63	0.77	0.41	0.58
Setting3.3	1.00	1.00	1.00	0.70	0.60	0.72	0.52	0.38	0.55	0.75	0.37	0.53

(c)	**(I)**			**(II)**			**(III)**			**(IV)**		
	R	***J***	***FM***	***R***	***J***	***FM***	***R***	***J***	***FM***	***R***	***J***	***FM***
Setting1.1	0.94	0.92	0.94	0.50	0.36	0.53	1.00	1.00	1.00	0.75	0.38	0.55
Setting1.2	0.93	0.90	0.93	0.51	0.37	0.54	0.61	0.46	0.63	0.65	0.23	0.38
Setting1.3	0.93	0.90	0.94	0.51	0.38	0.55	0.54	0.40	0.57	0.65	0.23	0.38
Setting2.1	1.00	1.00	1.00	0.50	0.36	0.53	1.00	0.99	1.00	1.00	0.99	0.99
Setting2.2	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.36	0.53	0.78	0.45	0.62
Setting2.3	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.36	0.53	0.76	0.40	0.57
Setting3.1	1.00	1.00	1.00	0.50	0.36	0.53	1.00	1.00	1.00	0.84	0.56	0.71
Setting3.2	1.00	1.00	1.00	0.56	0.43	0.59	0.59	0.44	0.61	0.75	0.37	0.54
Setting3.3	1.00	1.00	1.00	0.71	0.60	0.73	0.52	0.37	0.54	0.74	0.35	0.52

(d)	**(I)**			**(II)**			**(III)**			**(IV)**		
	R	***J***	***FM***	***R***	***J***	***FM***	***R***	***J***	***FM***	***R***	***J***	***FM***
Setting1.1	0.67	0.58	0.71	0.50	0.36	0.53	1.00	0.99	1.00	0.75	0.37	0.54
Setting1.2	0.68	0.58	0.71	0.51	0.38	0.55	0.60	0.46	0.63	0.64	0.22	0.37
Setting1.3	0.68	0.59	0.71	0.51	0.37	0.54	0.55	0.40	0.57	0.64	0.22	0.35
Setting2.1	1.00	1.00	1.00	0.50	0.36	0.53	0.99	0.99	0.99	1.00	0.99	0.99
Setting2.2	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.36	0.53	0.78	0.44	0.61
Setting2.3	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.36	0.53	0.77	0.41	0.58
Setting3.1	0.99	0.98	0.99	0.50	0.36	0.53	1.00	1.00	1.00	0.84	0.57	0.72
Setting3.2	0.99	0.98	0.99	0.57	0.44	0.61	0.59	0.45	0.62	0.73	0.35	0.52
Setting3.3	0.99	0.99	0.99	0.66	0.55	0.69	0.52	0.38	0.55	0.72	0.34	0.50

5 Conclusion

In this paper a procedure for clustering compositional data trajectories has been proposed. The procedure take advantage of the functional nature of observed data and make use of spline smoothing to separate “true values” from random noise and measurement errors. Moreover spline smoothing techniques are particularly suitable in this context since they allow comparison among trajectories measured on a misaligned grid. This comparison can be done by realigning observed data on a regular grid via prediction.

The most critical issues in cluster analysis are represented by the clustering algorithm and by the metric used to build the dissimilarity matrix. In the presented simulation study we found that the clustering algorithm is far less important than the employed metric. Two possible metrics are proposed. The first consider both the center of the observed trajectories and their shapes, while the second focus on the shape only.

Of course the proposed metrics are just two out of a whole series of possible instruments which can be used to measure trajectory dissimilarity. A number of different metrics could be built in order to take into account different features in the compositional trajectories. Each metric would result in a different clustering structure, and such difference can be ascribed to the different focus the researcher justifying the use of such metrics.

Acknowledgements

The research leading to the writing of this paper was funded by a 2006 grant (project no. 2006139812.003, sector: Economics and Statistics) for research projects of national interest, by the Italian Ministry of the University and Scientific and Technological Research.

References

Abraham, C., Cornillon, P.A., Matzner-Løber, E., Molinari, N. (2003). Unsupervised Curve Clustering using B-Splines. *Scandinavian Journal of Statistics 30*(3), 581-595.

Aitchison, J. (1982). The statistical analysis of compositional data (with discussion). J. R. Statist. Soc. B., 44, 139-177.

Aitchison, J. (1986). *The Statistical Analysis of Compositional Data*. New York: Chapman & Hall.

Aitchison, J. (1992). On Criteria for Measures of Compositional Difference. *Mathematical Geology 24*(4), 365-379.

Billheimer, D., Guttorp, P., Fagan, W. (2001). Statistical Interpretation of Species Composition. *Journal of the American Statistical Association*, 96, 1205-1214.

Ferrero, L., Bolzacchini, E., Perrone, M.G., Petraccone, S., Sangiorgi, G., Lo Porto, C., Ferrini, B.S., Lazzati, Z., Riccio, A., Previtali, E., Clemenza, M., Bruno, F., Cocchi, D., Greco, F. (2007) Influence of the Mixing Layer on the concentration and size distribution of Particulate Matter over Milan. *EAC2007, Salzburg, http://www.gaef.de/EAC2007/.*

Ghigo, S., Giovenali, E., Ignaccolo, R. (2006). Functional Cluster Analysis per l’ottimizzazione della rete di monitoraggio della qualità dell’aria in Piemonte. *GRASPA Working paper* n.26.

Guha, S., Rastogi, R., Shim, K. (1998). CURE: An Efficient Clustering Algorithm for Large Databases. In: *Proceedings of the ACM SIGMOD Conference*.

Halkidi, M., Vazirgiannis, M., Batistakis, I. (2000). Quality Scheme Assessment in the Clustering Process. In *Proceedings of PKDD*, Lyon, France.

Halkidi, M., Batistakis, Y., Vazirgiannis, M. (2001). On Clustering Validation Techniques. *Journal of Intelligent Information Systems 17*, 107–145.

Hartigan, J. A. (1975). *Clustering algorithms*. New York: Wiley.

Hastie, T., Tibshirani, R., Friedman, J. H. (2001). *The Elements of Statistical Learning*. New York: Springer.

Jain, A.K., Murty, M.N., Flyn, P.J. (1999). Data Clustering: A Review. *ACM Computing Surveys*, 31, 264–323.

Ludwig, F.L., Jiang, J., Chen, J. (1995). Classification of ozone and heather patterns associated with high ozone concentrations in the San Francisco and Monterey Bay areas. *Atmospheric Environment*, 29, 2915–2928.

Ramsay, J.O., Silverman, B. (2005). *Functional data analysis*. Second Edition. Springer-Verlag.

Theodoridis, S., Koutroubas, K. (1999). *Pattern Recognition*. Academic Press.