Statistical Methods & Applications (2023) 32:1123–1140
https://doi.org/10.1007/s10260-022-00665-y

ORIGINAL PAPER

Boosted-oriented probabilistic smoothing-spline clustering of series

Carmela Iorio[1] · Gianluca Frasso[2] · Antonio D'Ambrosio[1] · Roberta Siciliano[3]

Accepted: 9 October 2022 / Published online: 27 October 2022

Abstract

Fuzzy clustering methods allow the objects to belong to several clusters simultaneously, with different degrees of membership. However, a factor that influences the performance of fuzzy algorithms is the value of fuzzifier parameter. In this paper, we propose a fuzzy clustering procedure for data (time) series that does not depend on the definition of a fuzzifier parameter. It comes from two approaches, theoretically motivated for unsupervised and supervised classification cases, respectively. The first is the Probabilistic Distance clustering procedure. The second is the well known Boosting philosophy. Our idea is to adopt a boosting prospective for unsupervised learning problems, in particular we face with non hierarchical clustering problems. The global performance of the proposed method is investigated by various experiments.

Keywords Boosting · Fuzzy clustering · Probabilistic distance clustering · Cluster validity

✉ Carmela Iorio
carmela.iorio@unina.it

Gianluca Frasso
gianluca.frasso@wur.nl

Antonio D'Ambrosio
antdambr@unina.it

Roberta Siciliano
roberta@unina.it

[1] Department of Economics and Statistics, University of Naples Federico II, Via Cinthia, M.te S. Angelo, 80126 Naples, Italy

[2] Wageningen Food Safety Research, Wageningen University and Research, Akkermaalsbos, 2, 6708WB Wageningen, The Netherlands

[3] Department of Electrical and Information Technology, University of Naples Federico II, Via Claudio, 21, 80125 Naples, Italy

1 Introduction

We propose a fuzzy approach for clustering data (time) series. The goal of clustering is to discover groups so that objects within a cluster have high similarity among them, and at the same time they are dissimilar to objects in other clusters. Many clustering algorithms for time series have been introduced in the literature. Since clusters can formally be seen as subsets of the data set, one possible classification of clustering methods can be according to whether the subsets are fuzzy (soft) or crisp (hard). Let $\mathcal{D}$ be a data set consisting of N series $\{y_1, y_2, ..., y_N\} \subset \mathbb{R}^n$ and let K be an integer, with $2 \leq K < N$, the goal is to partition $\mathcal{D}$ into $\mathcal{C}_K$ groups. Crisp clustering methods are based on classical set theory, and restrict that each object of data set belongs to exactly one cluster. It means partitioning the data $\mathcal{D}$ into a specified number of mutually exclusive clusters $\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_K$.

The idea of fuzzy set was conceveid by Zadeh (1965). Fuzzy clustering methods do not assign objects to a cluster but suggest degrees of membership to each group. The larger is the value of the membership value for a given object with respect to a cluster, the larger is the probability of that object to be assigned to that cluster. Several clustering criteria have been proposed to identify fuzzy partition in $\mathcal{D}$. Among these proposals, the most popular method is fuzzy c-means.

Proposed by Dunn (1973) and developed by Bezdek (1981), fuzzy c-means considers each data point as a possible member of multiple clusters with a membership value. This algorithm is based on minimization of the following objective function:

$$J_m = \sum_{i=1}^{N} \sum_{k=1}^{K} (\mu_{ik})^m \|y_i - c_k\|^2 \tag{1}$$

s.t.

$$\begin{array}{l} \mu_{ik} \in [0,1], \ \forall i,k; \\ \sum_{k=1}^{K} \mu_{ik} = 1; \\ 0 < \sum_{i=1}^{N} \mu_{ik} < N. \end{array}$$

In the Eq. (1), m is any real number greater than 1, μ_{ik} is the degree of membership of y_i in the cluster k and $\|\cdot\|$ is any norm expressing the similarity between any measured data and the center. The parameter m is called *fuzzifier* or *weighting coefficient*. To perform fuzzy partitioning, the number of clusters and the weighting coefficient have to be choosen. The procedure is carried out through an iterative optimization of the objective function shown above, with the update of membership value μ_{ik} and the cluster centers c_k by solving:

$$c_k = \frac{\sum_{i=1}^{N} (\mu_{ik})^m y_i}{\sum_{i=1}^{N} (\mu_{ik})^m}, \ k = 1, \ldots, K.$$

$$\mu_{ik} = \left(\sum_{h=1}^{K} \left(\frac{\sqrt[(m-1)]{\|y_i - c_k\|^2}}{\sqrt[(m-1)]{\|y_i - c_h\|^2}} \right) \right)^{-1} \ i = 1, \ldots, N; k = 1, \ldots, K.$$

The loop will stop when

$$max_{ik}|\mu_{ik}^{(l+1)} - \mu_{ik}^{(l)}| < \varepsilon,$$

where ε is a small number for stopping the iterative procedure, and l indicates the iteration steps.

One of limitations of fuzzy c-means clustering is the value of fuzzifier m. A large fuzzifier value tends to mask outliers in data sets, i.e. the larger m, the more clusters share their objects and viceversa. For $m \to \infty$ all data objects have identical membership to each cluster, for $m = 1$, the method becomes equivalent to k-means. The role of the weighting exponent has been well investigated in literature.

Pal and Bezdek (1995) suggested taking $m \in [1.5, 2.5]$. Dembélé and Kastner (2003) obtain the fuzzifier with an empirical method calculating the coefficient of variation of a function of the distances between all objects of the entire dataset. Yu et al. (2004) proposed a theoretical upper bound for m that can prevent the sample mean from being the unique optimizer of a fuzzy c-means objective functions. Futschik and Carlisle (2005) search for a minimal fuzzifier value for which the cluster analysis of the randomized data set produces no meaningful results, by comparing a modified partitions coefficient for different values of both parameters. Schwämmle and Jensen (2010) showed that the optimal fuzzfier takes values far from the its frequently used value equal to 2. The authors introduced a method to determine the value of the fuzzifier without using the current working data set. Then for high dimensional ones, the fuzzifier value depends directly on the dimension of data set and its number of objects. For low dimensional data set with small number of objects, the authors reduce the search space to find the optimal value of the fuzzifier. According to the authors, this improvement helps choosing the right parameter and saving computational time when processing large data set. In the robust learning-based algorithm proposed by Yang and Nataliani (2017) the same value of 2 is chosen for the weighting coefficient. On the basis of a robust selection analysis of the algorithm, Wu (2012) founds that a large value of m will make fuzzy c-means algorithm more robust to noise and outliers. The author suggested to use value of the fuzzifier ranging between 1.5 and 4. By exploiting the quantum concept, Patel et al. (2015) proposed an evolutionary fuzzy c-means where the value of fuzzification index is represented in terms of quantum bits. Dotto et al. (2017) for choosing the fuzzification parameter monitor simultaneously the proportion of hard assignments and the relative entropy of fuzzy weights. By plotting the proposed procedure for different values of m, the user can set the degree of fuzzification by considering a compromise between the above mentioned quantities. Within the framework of robust clustering presented in Cerioli et al. (2018), Farcomeni and Dotto (2018) generalized Average Within-Cluster Distance to fuzzy clustering by setting a grid of values for the parameters and choosing $m = 1.3$. In order to increase efficiency of deriving a valid fuzzifier value, Cho (2022) introduced the Interval type-2 possibilistic fuzzy C-means in which suitable fuzzifier values for each data are obtained by an algorithm that includes the analysis of histogram and the Gaussian Curve Fitting method.

Since the weighting coefficient determines the fuzziness of the resulting classification, we propose a method that is independent from the choice of the fuzzifier. It comes from two approaches, theoretically motivated for unsupervised and supervised classification cases respectively. The first is the Probabilistic Distance (PD) clustering procedure defined by Ben-Israel and Iyigun (2008). The second is the well known Boosting philosophy. From the PD approach we took the idea of determining the probabilities of each series to any of the k clusters. As this probability is unequivocally related to the distance of each series from the centers, there are no degrees of freedom in determine the membership matrix. From the Boosting approach (Freund and Schapire 1997) we took the idea of weighting each series according some measure of badness of fit in order to define an unsupervised learning process based on a weighted re-sampling procedure. As a learner for the boosting procedure we use a smoothing spline approach. Among the smoothing spline techniques, we chose the penalized spline approach (Eilers and Marx 1996) because of its flexibility and computational efficiency. This paper is organized as follows: Sect. 2 contains our proposal, in Sect. 3 the results of some experimental evaluation studies are carried out and some concluding remarks are presented in Sect. 4.

2 Boosted-oriented probabilistic clustering of time series

2.1 The key idea

The boosting approach is based on the idea that a supervised learning algorithm (weak learner) improves its performance by learning from its errors (Freund and Schapire 1997). It consists of an ensemble method that works with a resampling procedure (Dietterich 2000). Our idea is to adapt the boosting philosophy to unsupervised learning problems, specially to non hierarchical cluster analysis. In such a case there not exists a target variable, but as the goal is to assign each instance (i.e. a series) to a cluster, we have a target instance. In other words, we switch from a target variable to a target instance point of view. We take each cluster center as a representative instance for each series and we assume as a synthetic index of the global performance a loss function to be minimized. The probability of each instance to belong to a given cluster is assumed to be the individual contribution of a given instance to the overall solution. In contrast to the boosting approach, the larger the probability of a given series to be member of a given cluster, the larger the weight of that series in the resampling process. As a learner either a smoothing spline techniques or a regression model can be used. We decided to use a penalized spline smoother because of its flexibility and computational efficiency. To define the probabilities of each series to belong to a given cluster we use the PD clustering approach (Ben-Israel and Iyigun 2008). This approach allows us to define a suitable loss function and, at the same time, to propose a fuzzy clustering procedure that does not depend on the definition of a fuzzifier parameter.

2.2 P-splines smoothing

Suppose we observe a set of data $\{x, y\}_{t=1}^{n}$, where the vector x indicates the independent variable (e.g. time) and y is modeled as $y = \mu + \epsilon$ where ϵ is a random error term and μ is a smooth signal. We can estimate the mean function using penalized splines (or simply P-splines) (Eilers and Marx 1996). P-splines are flexible smoothers combining B-spline bases (De Boor 1978) and discrete difference penalty operators. In order to estimate μ, we need to solve of the penalized least squares problem

$$\underset{a}{\text{argmin}}\ S(a) = \|y - \mu(a)\|^2 + Pen(a)\ = \|y - Ba\|^2 + \lambda\|D^{(d)}a\|^2$$

where B is a B-spline basis matrix built on a generous set of equally spaced internal knots (Eilers and Marx 2010), a is a vector of spline coefficients and $D^{(d)}$ is $d-$order difference operator (usual choices are $d = 2$ or 3). The regularization parameter λ tunes the amount of smoothing applied to the final fit (for $\lambda \to \infty$ the estimates tend to be constant while for $\lambda \to 0$ the smoother tends to interpolate the observations). The parameter λ must be selected using a suitable procedure. Common criteria are, the Akaike Information Criterion (AIC) and the (generalized) cross validation method. Here, we adopt the V-curve method proposed by Frasso and Eilers (2015). Differently from the aforementioned alternatives, the V-curve is computationally efficient (because does not require the computation of the effective degrees of freedom) and ensures a more robust fit against possible noise serial correlation.

2.3 PD clustering approach

Let $\mathcal{D}$ be a dataset consisting of N series $\{y_1, y_2, ..., y_N\} \subset \mathbb{R}^n$ and let $\mathcal{C}_k$ be $k - th$ cluster, with $k \in (1, K)$, partitioning $\mathcal{D}$. We suppose that each series has the same domain of length n. If the time series included in the sample are not aligned (i.e., the start and end point of the domain are not the same) we need a pre-processing step. To this end, we suggest to adopt the parametric time warping (PTW) framework (Eilers 2004).

At each cluster $\mathcal{C}_k$ is associated a cluster center c_k, with $k = 1, ..., K$.

Let $d_{i,k} = d(y_i, c_k)$ be a distance function of the $i - th$ series from the $k - th$ cluster center.

Let $P_{i,k} = P(y_i, \mathcal{C}_k)$ be the probability of the $i - th$ series belonging to the $k - th$ cluster.

For each series $y \in \mathcal{D}$ and each cluster $\mathcal{C}_k$, we assume the following relation between probabilities and distances (Ben-Israel and Iyigun 2008):

$$P_{i,k} d_{i,k} = constant(i). \tag{2}$$

The constant in (2) only depends on series y and it is independent of the cluster k. Equation (2) allows to define the membership probabilities as (Heiser 2004; Ben-Israel and Iyigun 2008):

$$P_{i,k} = \frac{\prod_{j \neq k} d_{i,j}}{\sum_{k=1}^{K} \prod_{j \neq k} d_{i,j}} \tag{3}$$

2.4 The algorithm

Since the probabilities as defined in Eq. (3) sum up to one among the clusters, we use the quantity $\prod_{k=1}^{K} P_{i,k}$ as a measure of compliance representation of the $i-th$ series with respect to the overall solution of the clustering procedure. It is easy to note that $\prod_{k=1}^{K} P_{i,k} = 0$ if the $i-th$ series exactly matches with one of the K cluster centers, as well as $\prod_{k=1}^{K} P_{i,k} = K^{-K}$ if there is maximum uncertainty in assigning the $i-th$ series to any cluster center. For this reason, to measure the clustering compliance solution, we adopted a Badness of Clustering (BC) index defined as follows:

$$BC = \frac{1}{N} \sum_{i=1}^{N} \left(\prod_{k=1}^{K} P_{i,k} \right) K^{K}. \tag{4}$$

Equation (4) is a synthetic uncertainty clustering measure: the lower its value, the better the solution. It equals zero when there is a perfect solution (i.e., each series has probability equal to one to belong to some cluster center). The maximum possible value of Eq. (4) is 1, when each series has probability equal to K^{-1} to belong to each of the K cluster. The BC index allows to compare the overall clustering solution when the number K of the clusters differs.

From Eq. (4) we define the following loss function to be minimized as

$$\beta = \sum_{i=1}^{N} \left(\prod_{k=1}^{K} P_{i,k} \right) K^{K}. \tag{5}$$

Let $\gamma_{i,k} = d_{i,k} / \max_{k=1}^{K} d_{i,k}$ be the contribution of the $i-th$ series to generate the $k-th$ cluster.

Let $\mathbf{\Gamma}$ be a $N \times K$ indicator matrix whose entries are 1 if $P_{i,k} > P_{i,h}$ $(k, h = 1, \ldots, K,$ $k \neq h)$ and -1 otherwise.

We define the weight of the $i-th$ series for the $k-th$ cluster as

$$w_{i,k} = \beta^{\gamma_{i,k} \Gamma_{i,k}}.$$

For each cluster k, the weights are first normalized in this way:

$$w_{i,k}^{\cdot} = \frac{w_{i,k}}{\sum_{h=1}^{K} w_{i,h}},$$

then within each cluster we set

$$W_{i,k} = \frac{w^{\cdot}_{i,k}}{\sum_{i=1}^{N} w^{\cdot}_{i,k}}. \tag{6}$$

For each cluster k, a sample $\mathcal{L}^{(k)}$ is extracted with replacement from $\mathcal{D}$, taking in account Eq. (6). Then the cluster centers $\widehat{c}_k = B\widehat{a}$, $k = 1, \ldots, K$ are estimated by using a P-spline smoother. These centers are then used to compute the membership probabilities according to Eq. (3) for the next iteration. The cluster centers are re-estimated and adaptively updated with an optimal spline smoother.

The choice of the metric depends on the nature of the series, the optimal P-spline smoothing procedure frames our approach in the class of model-based clustering techniques but any suitable smoother can be adopted. Box 1 shows the pseudo-code of our the Boosted-Oriented Smoothing Spline Probabilistic Clustering algorithm.

Box 1 Boosted-oriented smoothing-spline probabilistic clustering of time series

input $\mathcal{D}$
initialize: maxiter = maximum number of iterations; K = the number of clusters; d = a suitable distance measure; c_k, $k = 1 : \ldots, K$ random cluster centers.
for iter=1:maxiter **do**
- compute the $N \times K$ distance matrix $D = [d_{i,k}]$ $\forall i, k$;
- compute the membership probabilities $P = [P_{i,k}]$ $\forall i, k$ as in equation (3);
- compute $\beta^{[iter]}$ as in equation (5);
- assign the weights to each series for each cluster and compute the $N \times K$ matrix W as in equation (6);
for $k = 1 : K$ **do**
- extract the sample $\mathcal{L}^k$ from $\mathcal{D}$
- compute center $\widehat{c}_k^{[iter]} = B\widehat{a}_k$
end for
if iter= 1 **then**
- $\widehat{c}_k^* = B\widehat{a}_k$
else
for $k = 1 : K$ **do**
- update cluster centers $\widehat{c}_k^* = B\widehat{a}_k^*$, with $\widehat{a}_k^* = (B^\top B + \lambda D^\top D)^{-1} B^\top \widehat{c}_k^{[1:iter]}$
end for
end if
end for
output: estimated cluster centers $\widehat{c}_k^*$, membership probabilities matrix P.

The procedure described in Box 1 is repeated a certain number of time due to the sensitivity of final solution to the random choice of cluster center.

3 Experimental evaluation

To evaluate the performance of the proposed algorithm, we conducted three experiments. In estimating the optimal P-splines smoother, always we used the V-curve criterion to select the optimal λ parameter, and we used a number of interior knots equal to $\min(\frac{n}{4};40)$, in which n is the length of time domain, as suggested by Ruppert (2002). As a measure of goodness of fuzzy partitions, we use the Adjusted Concordance Index (ACI) proposed by D'Ambrosio et al. (2021), which is the fuzzy extension of the Adjusted Rand Index (Hubert and Arabie 1985).

As true fuzzy partition, we always computed the true cluster centers with an optimal P-spline smoother, and then we computed the true probabilities by applying Eq. (3).

3.1 Simulated data

As a first experiment, we generated $K = 6$ clusters of numerical series at $n = 10$ equally spaced time points in [0, 1] as described in Coffey et al. (2014). Distinct cluster specific models were used (subscript i refers to the series, subscript j refers to the time domain):

$$
\begin{aligned}
y_{ij}^{(1)} &= \alpha_i + \sin(\beta_i \times \pi \times x_{ij}) + \gamma_i + \varepsilon_{ij} \\
y_{ij}^{(2)} &= x_{ij} + (\delta_i)^{-3} + \iota_i + \gamma_i + \varepsilon_{ij} \\
y_{ij}^{(3)} &= \nu_i + \gamma_i + \varepsilon_{ij} \\
y_{ij}^{(4)} &= \zeta_i + \cos(\zeta_i \times \pi \times x_{ij}) + \gamma_i + \varepsilon_{ij} \\
y_{ij}^{(5)} &= \xi_i - \eta_i \times \exp(-\theta_i \times x_i) + \gamma_i + \varepsilon_{ij} \\
y_{ij}^{(6)} &= -3(x_{ij} - 0.5) + \gamma_i + \varepsilon_{ij}
\end{aligned}
$$

where:

$\alpha_i \sim N(\sqrt{2};\sigma_e^2)$ with $\sigma_e^2 = 0.08$, $\beta_i \sim N(4 * \pi;\sigma_e^2)$, $\delta_i \sim N(0.75;\sigma_e^2)$, $\iota_i \sim N(1;\sigma_e^2)$, $\nu_i \sim N(0;\sigma_e^2)$, $\zeta_i \sim N(2;\sigma_e^2)$, $\xi_i \sim N(2;\sigma_v^2)$ with $\sigma_v^2 = 0.85$, $\eta_i \sim N(4;\sigma_v^2)$, $\theta_i \sim N(6;\sigma_e^2)$, $\gamma_i \sim N(0;\sigma_u^2)$ with σ_u^2 ranging from 0.3 to 1 and ε_{ij} is an autoregressive model of order 1.

Cluster means were chosen to reflect the situation where there are series that show little variation in value over time (as given by cluster 3) and series which have distinct signal over time. Cluster sizes were equal to 90, 50, 100, 25, 60 and 35, for cluster 1, 2, 3, 4, 5, 6 respectively, giving a total number of 360 simulated series. Data set is plotted in Fig.1.

Given the nature of the simulated series, we are interested in the similarity of the shape of the series. For this reason the chosen metric was the Penrose shape distance (Penrose 1952), defined as:

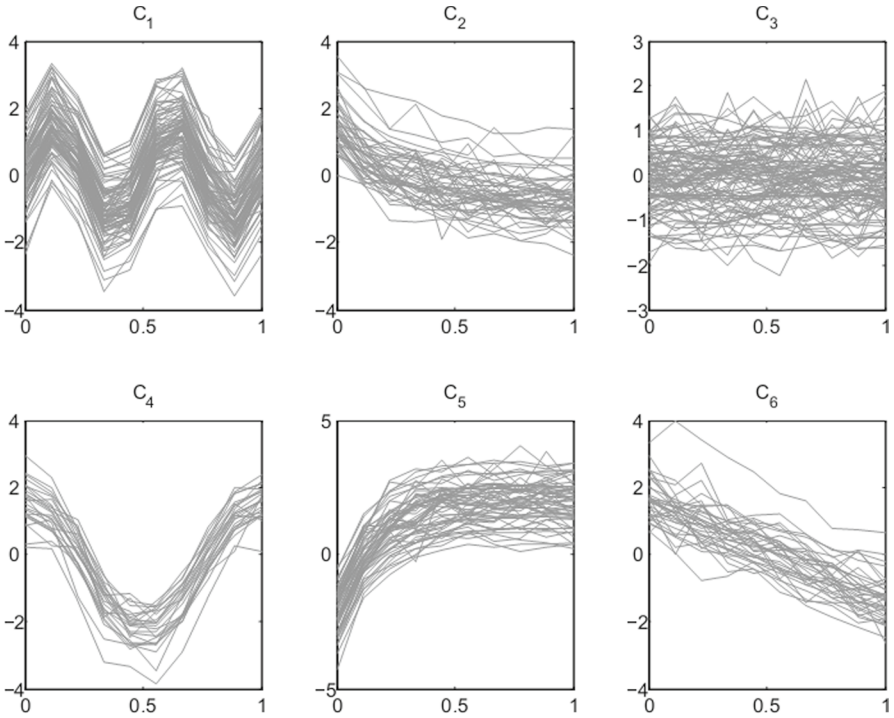

Fig. 1 Data set generated for simulation study

$$d_{i,j} = \sqrt{\frac{n_i}{n_i - 1}\left(d_{i,j}^2 - q_{ij}^2\right)}, \tag{7}$$

where $d_{i,j}^2$ is the squared average Euclidean distance coefficient and $q_{ij}^2 = \frac{1}{n_i^2}\left(\sum_{j=1}^{n_i} y_{ji} - \sum_{j=1}^{n_i} c_{jk}\right)^2$.

We performed five analysis with 100, 500, 1000, 5000 and 10000 boosting iterations. In all cases we set 10 random starting points. Figure 2 shows the behavior of the BC function as defined in Eq. (4) during the boosting iterations. In this case the *BC* values appear to be non-increasing as the number of iterations increases. The values of the *BC* function are equal to 0.3615, 0.2783, 0.2643, 0.2 584, 0.2583 for 100, 500, 1000, 5000 and 10000 boosting iterations respectively. All the solutions return in fact the same results in terms of estimated centers: in example, Fig. 3 shows the estimated cluster centers for each cluster as returned by the first analysis.

For this data set, by using the Penrose shape distance, the ACI is equal to 0.8599, 0.8954, 0.9059, 0.9178 and 0.9194 for the solutions with respectively 100, 500, 1000, 5000 and 10000 boosting iterations. Even if the solutions in terms of "hard" clustering are the same, the difference in terms of adjusted concordance index indicates that the partitions returned by the proposed algorithm are really close to the true one. The true value of the *BC* index is 0.1977.

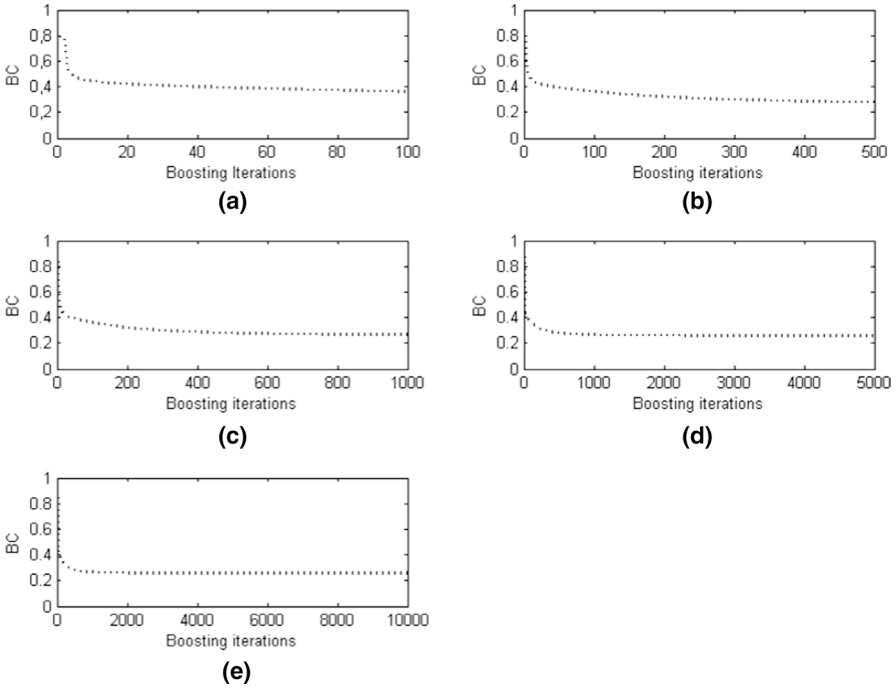

Fig. 2 *BC* function progress through: **a** = 100 boosting iterations; **b** = 500 boosting iterations; **c** = 1000 boosting iterations; **d** = 5000 boosting iterations; **e** = 10000 boosting iterations

3.2 Synthetic data set

synthetic.tseries data set is freely available from the `TSclust` R-package Montero and Vilar (2014). synthetic.tseries data consist of three partial realizations of length $n = 200$ of six first order autoregressive models. Figure 4 shows separately the six groups of series.

Subplot (a) shows an AR(1) process with moderate autocorrelation. Subplot (b) contains series from a bi-linear process with approximately quadratic conditional mean. Subplot (c) is formed by an exponential autoregressive model with a more complex non-linear structure. Subplot (d) shows a self-exciting threshold autoregressive model with a relatively strong non-linearity. Subplot (e) contains series generated by a general non-linear autoregressive model and subplot (f) shows a smooth transition autoregressive model presenting a weak non-linear structure. As we did not generated these series we do not show completely the simulation setting. For more details about the generating models we refer to (Montero and Vilar 2014, p. 24).

Assuming that the aim of cluster analysis is to discover the similarity between underlying models, the "true" cluster solution is given by the six clusters involving the three series from the same generating model. Given the nature of the data set considered, we use a periodogram-based distance measure proposed by

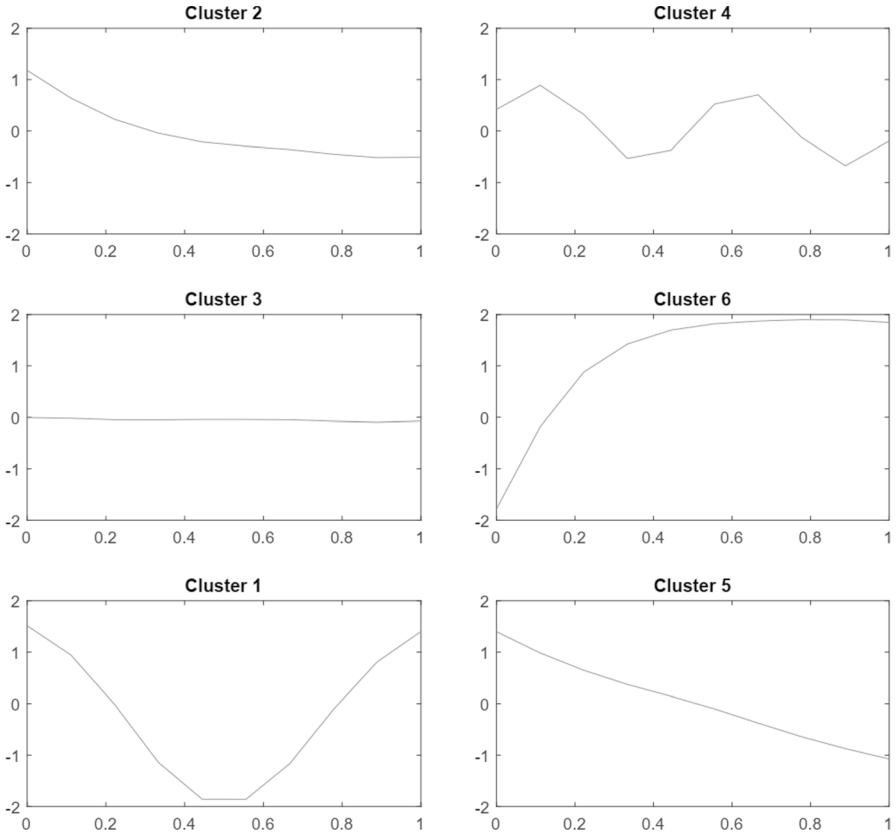

Fig. 3 Simulated data: recognized bari-center

Caiado et al. (2006). It assesses the dissimilarity between the corresponding spectral representation of time series.

By following also the suggestion of Montero and Vilar (2014), an interesting alternative to measure the dissimilarity between time series is the frequency domain approach. Power spectrum analysis is concerned with the distribution of the signal power in the frequency domain. The power-spectral density is defined as the Fourier transform of the autocorrelation function of $i-th$ series. It is a measure of self-similarity of a signal with its delayed version. The classic method for the estimation of the power spectral density of an n-sample record is the periodogram introduced by Schuster (1897).

Let y and $y^{'}$ be two time series of length n.

Let $f_j = 2\pi j/n,\ j = 1, \ldots, n/2$ in the range 0 to π, be the frequencies of the series.

Let $PSD_y(f_j) = \frac{1}{n}\sum_{t=1}^{n} |y_t(f_j)\exp(-\imath t f_j)|^2$ and $PSD_{y'}(f_j) = \frac{1}{n}\sum_{t=1}^{n} |y_t'(f_j)\exp(-\imath t f_j)|^2$ be the periodograms of series y and $y^{'}$, respectively.

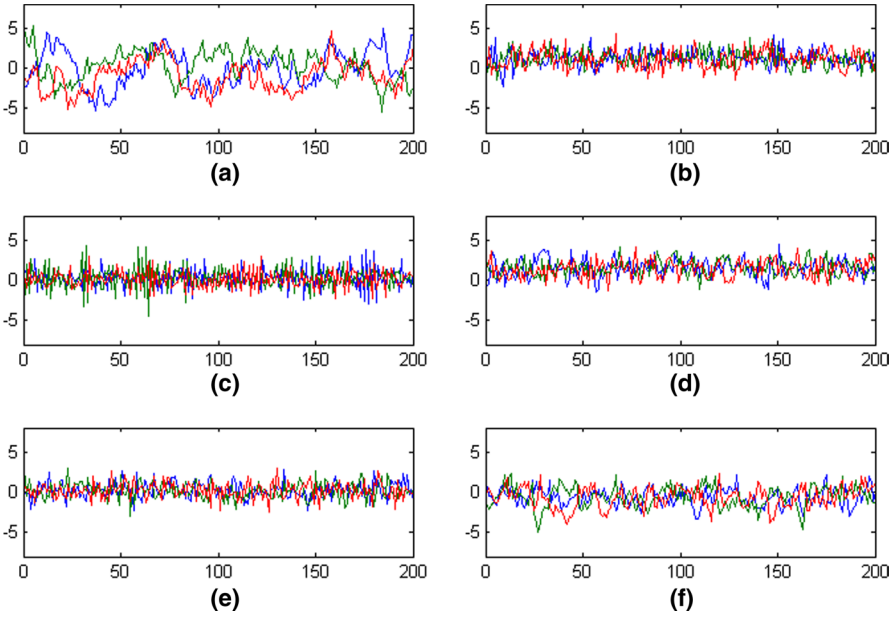

Fig. 4 Synthetic.tseries data set (color figure online)

Finally, the dissimilarity measure between y and y' proposed by Caiado et al. (2006) is defined as the Euclidean distance between periodogram ordinates:

$$d_{y,y'} = \sqrt{\sum_{j=1}^{(n/2)} [PSD_y(f_j) - PSD_{y'}(f_j)]^2}. \quad (8)$$

We performed our analysis by setting 800 boosting iterations and 10 random starting points.

Table 1 shows the results of applying our algorithm to the synthetic.tseries data set. Each series is assigned to the estimated cluster according to the value of the membership probability matrix (i.e., the largest membership probability value). In order to obtain the ACI, we computed the true cluster centers with a periodogram

Table 1 Confusion matrix from clustering on synthetic.tseries data set

		Estimated clusters					
		C1	C2	C3	C4	C5	C6
True clusters	a	0	0	0	0	0	3
	b	0	1	0	2	0	0
	c	3	0	0	0	0	0
	d	0	3	0	0	0	0
	e	0	0	3	0	0	0
	f	0	0	0	0	3	0

modeled by P-spline, then we computed the true probabilities by applying Eq. (3) by using the periodogram-based distance as in Eq. (8).

The ACI is equal to 0.9698. Even if the solutions in terms of "hard" clustering seems to be excellent (since only one series is misclassified), the difference in terms of ACI indicates that the partitions returned by the algorithm are really close to the true one.

3.3 A real data example

The "growth" data set is freely available from the internal repository of the R-package *fda* (Ramsay et al. 2010). This data set comes from the Berkeley Growth Study (Tuddenham 1954). Left hand side of Fig. 5 shows the growth curves of 93 children, 39 boys and 54 girls, starting by the age of one year till the age of 18. The right hand side of the same figure displays the corresponding growth velocities. In the framework of cluster analysis this data set was mainly used for problems of clustering of misaligned data (Sangalli et al. 2010; Vitelli et al. 2010). We performed two analysis with 800 boosting iterations and with 10 random starting points with $k = 2$. In the first partitioning analysis we used the Euclidean distance. The estimated centers of both the growth curves and the growth velocity curves are displayed respectively in the left and right hand side of Fig. 6. As it can be noted, Euclidean distance discriminates between children growing more and children growing less. This can be appreciated by looking at left hand side of the same figure. On average, as expected, boys grow more than girls. Nevertheless, Euclidean distance does not seem the right measure to be used in such a case. Probably researchers are interested in the shape of both growth and growth velocity curves during the years. For this reason, we repeated the analysis by using the Penrose shape distance as defined in

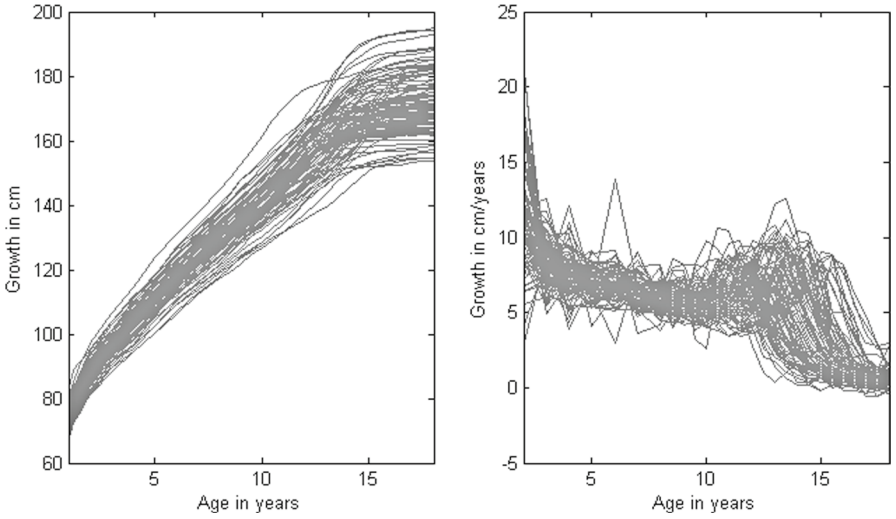

Fig. 5 Growth curves (left hand side) and growth velocity curves (right hand side) of 93 children from Berkeley Growth Study data

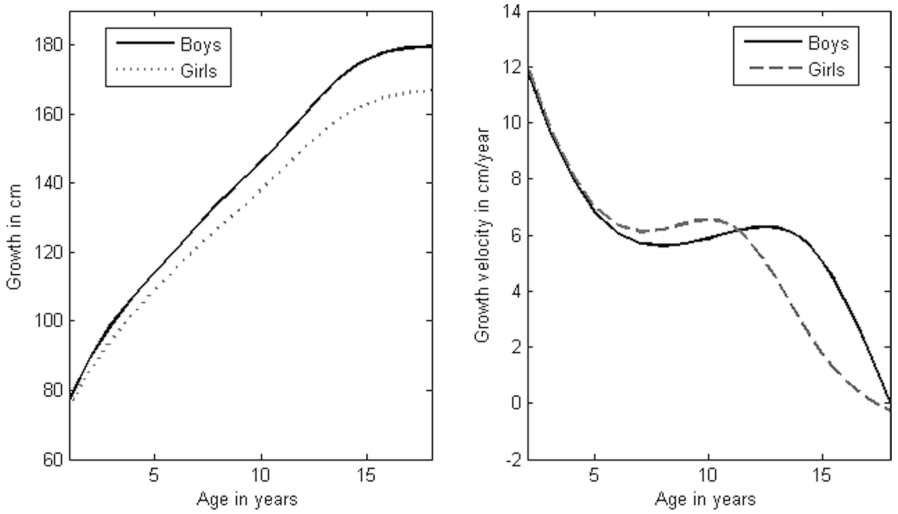

Fig. 6 Estimated centers of growth curves (left hand side) and growth velocities (right hand side): Euclidean distance

Eq. (7). Figure 7 shows the estimated centers for both the growth and the growth velocity curves. The recognized centers are really similar to the ones obtained by Sangalli et al. (2010) and Vitelli et al. (2010). Firstly, as confirmed by looking at tables 4 and 5 with respect to tables 2 and 3, there is a neat separation of boys and girls. Secondly, by looking at right hand side of Fig. 7, boys start to grow later but they seem to have a more pronounced growth, as it can be noticed by looking at the

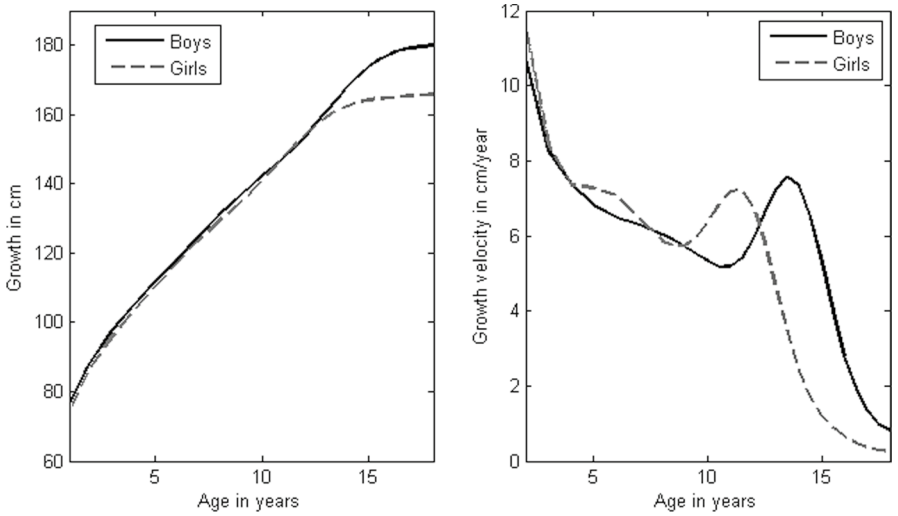

Fig. 7 Estimated centers of growth curves (left hand side) and growth velocities (right hand side): Penrose shape distance

Table 2 Confusion matrix of growth curves with the Euclidean distance. Series have been assigned to the clusters according the values of membership probabilities computed as in Eq. (3)

	Cluster 1	Cluster 2
Boys	23	16
Girls	16	38

Table 3 Confusion matrix of growth velocity curves with the Euclidean distance. Series have been assigned to the clusters according the values of membership probabilities computed as in Eq. (3)

	Cluster 1	Cluster 2
Boys	31	8
Girls	9	45

Table 4 Confusion matrix of growth curves with the Penrose shape distance. Series have been assigned to the clusters according the values of membership probabilities computed as in Eq. (3)

	Cluster 1	Cluster 2
Boys	0	39
Girls	52	2

Table 5 Confusion matrix of growth velocity curves with the Penrose shape distance. Series have been assigned to the clusters according the values of membership probabilities computed as in Eq. (3)

	Cluster 1	Cluster 2
Boys	36	3
Girls	4	49

higher peak in correspondence of 15 year. The ACI is equal to 0.8884 and 0.8240 by using the Euclidean distance for the partitions of growth and growth velocity curves respectively. The ACI is equal to 1.000 and 0.9246 by using the Penrose shape distance for the partitions of growth and growth velocity curves respectively.

4 Concluding remarks

In this paper we have presented a boosted-oriented probabilistic clustering of time series. Unlike the methods proposed so far in the literature, our methodology produces a final cluster that is independent of the choice of the fuzzifier. Our proposal merged two approaches, theoretically motivated for respectively unsupervised and

supervised classification cases, to propose a new non-hierachical fuzzy clustering algorithm. From the Probabilistic Distance (PD) clustering (Ben-Israel and Iyigun 2008) approach we shared the idea of determining the probabilities of each series to any of the k clusters. As this probability is directly related to the distance of each series from the cluster centers, there are no degrees of freedom in determine the membership matrix.

From the Boosting approach (Freund and Schapire 1997) we shared the idea of weighting each series according some measure of badness of fit in order to define an unsupervised learning process based on a weighted resampling procedure. In contrast to the boosting approach, the higher the probability of a given instance to be member of a given cluster, the higher the weight of that instance in the resampling process. As a learner we can use any smoothing spline technique. We used a P-spline smoother (Eilers and Marx 1996) because of its nice properties and we choose the optimal spline parameter with the V-curve criterion as defined by Frasso and Eilers (2015). In this way we defined a suitable loss function and, at the same time, we proposed a fuzzy clustering procedure that does not depend on the definition of a fuzzifier parameter.

To evaluate the performance of our proposal, we conducted three experiments, one of them on simulated data and the remaining two on data sets known in literature. The results show that our Boosted-oriented procedure show good performance in terms of data partitioning. Even if the final fuzzy partition is sensitive to the choice of a distance measure, it is independent on any other input parameters. This consideration allows to define a suitable true fuzzy partition with which evaluate the final solution in terms of Adjusted Concordance Index (D'Ambrosio et al. 2021). The weigthed re-sampling process allows each series to contribute to the composition of each cluster as well as the adaptive estimation of cluster centers allows the algorithm to learn by its progresses.

It is worth-nothing that, as in any partitioning problem, the choice of the distance measure can influence the goodness of partition.

Funding Open access funding provided by Università degli Studi di Napoli Federico II within the CRUI-CARE Agreement.

Declarations

Conflict of interest The Authors have declared no conflict of interest.

References

Ben-Israel A, Iyigun C (2008) Probabilistic d-clustering. J Classif 25(1):5–26

Bezdek JC (1981) Objective function clustering. Pattern recognition with fuzzy objective function algorithms. Springer, Berlin, pp 43–93

Caiado J, Crato N, Peña D (2006) A periodogram-based metric for time series classification. Comput Stat Data Anal 50(10):2668–2684

Cerioli A, Riani M, Atkinson AC, Corbellini A (2018) The power of monitoring: how to make the most of a contaminated multivariate sample. Stat Methods Appl 27(4):559–587

Cho J (2022) Data clustering for fuzzyfier value derivation. Volosencu C (Ed.), Fuzzy systems (chap. 7). IntechOpen

Coffey N, Hinde J, Holian E (2014) Clustering longitudinal profiles using p-splines and mixed effects models applied to time-course gene expression data. Comput Stat Data Anal 71:14–29

D'Ambrosio A, Amodio S, Iorio C, Pandolfo G, Siciliano R (2021) Adjusted concordance index: an extensionl of the adjusted rand index to fuzzy partitions. J Classif 38(1):112–128

De Boor C (1978) A practical guide to splines. Springer-Verlag, New York

Dembélé D, Kastner P (2003) Fuzzy c-means method for clustering microarray data. Bioinformatics 19(8):973–980

Dietterich TG (2000) Ensemble methods in machine learning. In: International workshop on multiple classifier systems, pp 1–15

Dotto F, Farcomeni A, García-Escudero LA, Mayo-Iscar A (2017) A fuzzy approach to robust regression clustering. Adv Data Anal Classif 11(4):691–710

Dunn JC (1973) A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. J Cyber 3(3):32–57

Eilers PH (2004) Parametric time warping. Anal Chem 76(2):404–411

Eilers PH, Marx BD (1996) Flexible smoothing with b -splines and penalties. Stat Sci 11(2):89–121

Eilers PH, Marx BD (2010) Splines, knots, and penalties. Wiley Interdiscip Rev Comput Stat 2(6):637–653

Farcomeni A, Dotto F (2018) The power of (extended) monitoring in robust clustering. Stat Methods Appl 27(4):651–660

Frasso G, Eilers PH (2015) L- and v-curves for optimal smoothing. Stat Modell 15(1):91–111

Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 55(1):119–139

Futschik ME, Carlisle B (2005) Noise-robust soft clustering of gene expression time-course data. J Bioinf Comput Biol 3(04):965–988

Heiser WJ (2004) Geometric representation of association between categories. Psychometrika 69(4):513–545

Hubert L, Arabie P (1985) Comparing partitions. J classif 2(1):193–218

Montero P, Vilar JA (2014) Tsclust: An r package for time series clustering. J Stat Softw 62(1):1–43

Pal NR, Bezdek JC (1995) On cluster validity for the fuzzy c-means model. IEEE Trans Fuzzy Syst 3(3):370–379

Patel OP, Bharill N, Tiwari A (2015) A quantum-inspired fuzzy based evolutionary algorithm for data clustering. In: 2015 IEEE international conference on fuzzy systems (fuzz-ieee) (pp. 1–8)

Penrose LS (1952) Distance, size and shape. Ann Eugen 17(1):337–343

Ramsay J, Wickham H, Graves S, Hooker G (2010) fda: Functional data analysis. r package version 2.2.6. http://CRAN.R-project.org/package=fda

Ruppert D (2002) Selecting the number of knots for penalized splines. J Comput Graph Stat 11(4):735–757

Sangalli ML, Secchi P, Vantini S, Vitelli V (2010) K-mean alignment for curve clustering. Comput Stat Data Anal 54(5):1219–1233

Schuster A (1897) On lunar and solar periodicities of earthquakes. Proc R Soc London 61(369–377):455–465

Schwämmle V, Jensen ON (2010) A simple and fast method to determine the parameters for fuzzy c-means cluster analysis. Bioinformatics 26(22):2841–2848

Tuddenham RD (1954) Physical growth of california boys and girls from birth to eighteen years. University of California publications in child development 1:183–364

Vitelli V, Sangalli LM, Secchi P, Vantini S (2010) Functional clustering and alignment methods with applications. Commun Appl Ind Math 1(1):205–224

Wu K-L (2012) Analysis of parameter selections for fuzzy c-means. Pattern Recogn 45(1):407–415

Yang M-S, Nataliani Y (2017) Robust-learning fuzzy c-means clustering algorithm with unknown number of clusters. Pattern Recogn 71:45–59
Yu J, Cheng Q, Huang H (2004) Analysis of the weighting exponent in the fcm. IEEE Trans Syst Man Cyber Part B (Cybernetics) 34(1):634–639
Zadeh LA (1965) Fuzzy sets. Inf Control 8(3):338–353