Information Sciences 624 (2023) 467–492

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Hard and soft clustering of categorical time series based on two novel distances with an application to biological sequences

Ángel López-Oriona [a,*], José A. Vilar [a], Pierpaolo D'Urso [b]

[a] Research Group MODES, Research Center for Information and Communication Technologies (CITIC), University of A Coruña, 15071 A Coruña, Spain
[b] Department of Social Sciences and Economics, Sapienza University of Rome, P. le Aldo Moro 5, Roma, Italy

ARTICLE INFO

Article history:
Received 19 August 2022
Received in revised form 20 December 2022
Accepted 22 December 2022
Available online 28 December 2022

Keywords:
Categorical time series
Association measures
Hard clustering
Fuzzy clustering
Biological sequences

ABSTRACT

Two novel distances between categorical time series are introduced. Both of them measure discrepancies between extracted features describing the underlying serial dependence patterns. One distance is based on well-known association measures, namely Cramer's v and Cohen's κ. The other one relies on the so-called binarization of a categorical process, which indicates the presence of each category by means of a canonical vector. Binarization is used to construct a set of innovative association measures which allow to identify different types of serial dependence. The metrics are used to perform crisp and fuzzy clustering of nominal series. The proposed approaches are able to group together series generated from similar stochastic processes, achieve accurate results with series coming from a broad range of models and are computationally efficient. Extensive simulation studies show that both hard and soft clustering algorithms outperform several alternative procedures proposed in the literature. Two applications involving biological sequences from different species highlight the usefulness of the introduced techniques.

1. Introduction

Clustering of time series concerns the challenge of splitting a set of unlabeled time series into homogeneous groups, which is a pivotal problem in many knowledge discovery tasks. Its applications encompass a broad spectrum of fields including artificial intelligence, computer science, biology, finance, environmental sciences, psychology, and medicine, among many others. As a result, time series clustering has attracted great attention in the data mining community, becoming extensively studied over the past few decades. Excellent reviews on the topic are provided in [1,2]. However, most of the works deal with continuous-valued time series, whereas cluster analysis of categorical series, the focus of this article, is usually neglected in the literature.

Categorical time series (CTS) are featured by taking values on a qualitative range consisting of a finite number of categories, which is referred to as *ordinal* range, if the categories exhibit a natural ordering, or *nominal* range, otherwise. In this work, the most general case of nominal range is considered. Indeed, dealing with unordered qualitative outcomes implies that some basic analytic tools are not longer applicable. Thus, standard measures of location (mean, median, quantiles), dispersion (standard deviation, range) and dependence (autocorrelation, partial autocorrelation) are not defined, and alternative measures considering the qualitative nature of the outcomes are needed to analyze CTS.

* Corresponding author.
E-mail addresses: oriona38@hotmail.com, a.oriona@udc.es (Á. López-Oriona), jose.vilarf@udc.es (J.A. Vilar), pierpaolo.durso@uniroma1.it (P. D'Urso).

https://doi.org/10.1016/j.ins.2022.12.065
0020-0255/

CTS arise in an extensive assortment of fields. Some illustrative examples are the stochastic modeling of DNA sequence data [3,4], the analysis of EEG sleep state scores [5], and the use of hidden Markov processes to model protein families [6]. A comprehensive overview of discrete-valued time series is provided by [7], including stochastic properties, modeling, insightful examples and practical implementation. In particular, many interesting applications can be addressed by grouping CTS. Typical examples are sequence alignment clustering, segmentation of customers based on their market baskets, clustering of patients according to the presented symptoms, and identifying web-user profiles according to the sequences of visited sites (commonly referred to as *clickstreams*).

Despite the wide range of applications, only a few works have addressed CTS clustering. [8] proposed a clustering algorithm based on a mixture of first order Markov models to group users with similar navigation behaviors. [9,10] introduced model-based procedures also relying on first order Markov Chains (MC) and allowing for covariates. [11] addressed the selection of the number of components for clustering based on mixtures of MC. In [12], a distance between Hidden Markov Models (HMM) characterizing the series was considered. Notice that [8–12] consider model-based procedures, i.e., assume specific models for the CTS subject to clustering, which makes their applicability to real databases rather limited. A more intuitive approach consists of introducing a distance between CTS to construct an initial pairwise dissimilarity matrix, and then applying a conventional clustering algorithm. However, it is challenging to define a proper distance between categorical data and usual distances (χ^2, Hamming, simple matching dissimilarity,...) ignore the underlying temporal structure. On the other hand, measuring dissimilarity between categorical sequences is a central problem in some specific fields. For instance, this is the case of the so-called sequence analysis [13], a topic of great interest in social sciences. In this framework, a set of variants of optimal matching (OM) algorithms and distances between inter-sequences have been provided [14–18]. Nevertheless, this kind of metrics aim at discriminating between shapes, but they are not designed to capture differences between the dynamic structures describing the global behavior of the series (e.g., by treating with stationary series). In [19], a different strategy was proposed to cluster clickstreams. Specifically, a dissimilarity measure combining both closeness of raw categorical values and similarity between dynamic behaviors was used as input to a modified version of the *K*-modes algorithm. A more detailed summary of the mentioned approaches, including some useful software libraries, is given in Tables 1 and 2 for the model-based and distance-based approaches, respectively.

Previous considerations support the need for further exploration of CTS clustering. In particular, the motivation behind this work is twofold. First, introducing model-free procedures considering the feature-based approach, a broadly used strategy to cluster continuous-valued time series but overlooked for categorical series. Secondly, developing fuzzy versions of the clustering algorithms. Some fuzzy algorithms have been proposed to cluster categorical data [22,23], but the adoption of the fuzzy logic in CTS clustering has not received proper attention. Note that, as in the case of numerical time series, the dependence structure of a categorical time series (e.g., a clickstream) may change over time so that it might belong to distinct clusters during different periods of time [24,25]. In addition, a fuzzy definition of the clusters allows to identify different underlying prototypes when the observed patterns do not differ too much from each other [24] (e.g., this might be the case for DNA sequences of viruses from the same family). In sum, introducing fuzziness in CTS clustering provides a desirable versatility to characterize the intrinsic clustering structure of the dataset. By accomplishing these two objectives, our work contributes to fill two important gaps concerning CTS clustering that currently exist in the literature.

In sum, this article is aimed at introducing crisp and fuzzy clustering algorithms for stationary CTS capable of: (i) grouping together categorical sequences generated from similar stochastic processes, (ii) achieving accurate results with series coming from a broad variety of categorical models, and (iii) performing the clustering task in low computation times.

To this aim, we first introduce an algorithm by considering the partitioning-based approach, where data objects are iteratively relocated between the clusters in such a way that the dispersion within clusters decreases at each iteration. The novelty of our procedure lies in how the dissimilarity between objects is measured. Since our target is to group series with similar underlying dependence structures, we propose to compare extracted features quantifying serial dependence. Unlike other works, we do not require determining a suitable metric between raw categorical data or assuming specific underlying models. In addition, our approach allows to overcome the noisy nature of the raw data, reduce dimensionality, compare series with different lengths or including missing data, and, more importantly, adjust the dissimilarity criterion to the specific application domain by selecting suitable features. The feature-based approach can be seen as a "universal" solution, more robust and usually less computationally intensive than other alternatives. The main challenge is to select proper features. Different choices have been considered to deal with real-valued time series, including autocorrelations [24], quantile dependence measures [26,25,27], frequency domain-based features [28,29], and the combined use of global features [30]. In the current setting, the selected features must take into account the nominal nature of the series. Specifically, we introduce two novel dissimilarities for categorical sequences. The first one combines the information provided by the elements defining two well-known association measures, namely the Cramer's v and the Cohen's κ. Both of them describe the serial dependence between categories in different ways and present attractive properties. The second metric arises from considering an alternative representation of the CTS through binary vectors taking the value one in the component associated with the observed category and zeros in the rest of components. This binarization process enables the computation of standard autocorrelations, which are used to define a new distance. Under stationarity, both distances are always well-defined and are intuitive and computationally efficient.

Next step consists of using the two proposed distances to construct novel clustering algorithms for CTS. We consider both the crisp and the fuzzy paradigms. In particular, when adopting the fuzzy approach, we simultaneously take advantage of

Table 1
Some references providing model-based approaches for clustering of CTS.

Paper	Method
Cadez et al. [8]	A mixture of first-order MC models is learned using the EM algorithm.
Dias [11]	The selection of the number of components for clustering based on finite mixtures of MC using several information criteria is addressed via a Monte Carlo study.
Pamminger and Frühwirth-Schnatter [9]	Two approaches based on a finite mixture of first-order MC models: (i) assuming that all series within a cluster are described by the same cluster-specific transition matrix, and (ii) assuming that the transition matrix of each time series deviates from an average group-specific transition matrix according to a Dirichlet distribution. Bayesian estimation using a two-block MC Monte Carlo sampler is considered.
Frühwirth-Schnatter et al. [10]	Algorithms in [9] are extended by formulating a probabilistic (logit type prior) model for the latent group indicators within the Bayesian classification rule by using a multinomial logit model.
Ghassempour et al. [12]	Each time series is characterized by a HMM and then the symmetrized Kullback–Leibler divergence between HMM is used to construct a distance matrix. The algorithm is valid for multivariate time series including both categorical and continuous variables.
Melnykov [20]	The R package **ClickClust** is introduced. This library is devoted to finite mixture modeling and model-based clustering of categorical sequences, with particular attention to the problem of grouping *clickstreams*. Methodological and algorithmic foundations of the package are discussed.

Table 2
Some references introducing distance measures between categorical sequences.

Paper	Method
Elzinga [14]	Four classes of alternatives to the optimal matching (OM) approach are introduced in order to compare categorical sequences. The proposed metrics are based on attributes of pairs of sequences, which are meaningful within the context of substantive social science theories.
Lesnard [15]	A specific OM algorithm (Dynamic Hamming Matching, DHM) is used to construct a dissimilarity matrix. DHM only employs substitution operations with time-dependent costs inversely proportional to transition frequencies. The behavior of DHM is compared to three classical OM variants (Hamming and Levenshtein I and II).
Halpin [16]	A variant of the OM algorithm (so-called OMv) based on weighting OM's elementary operations inversely with episode length.
Studer and Ritschard [17]	A comparative study of multiple ways of measuring dissimilarities between categorical sequences. The focus is put on differences concerning the order in which successive categories appear, the timing and the duration of the spells in successive categories. All metrics are available in the R package **TraMineR** [21].
Halpin [18]	The package **SADI** for STATA is introduced. **SADI** is devoted to sequence analysis including utilities as: dissimilarities between pairs of categorical sequences, graphical summaries and tools for cluster analysis, among others.
García-Magariños and Vilar [19]	A novel dissimilarity measure combining both closeness of raw categorical values and similarity between dynamic behaviors is used as input to a modified version of the *K*-modes algorithm.

both the discriminatory capability of the proposed distances and the assignment of gradual membership of the CTS to clusters. Extensive simulation studies involving a broad range of dependence models show the superiority of the proposed techniques with respect to other clustering algorithms employing alternative dissimilarities.

The remainder of the paper is organized as follows. Suitable features for measuring serial dependence in categorical sequences and two distances between CTS based on these features are introduced in Section 2. Crisp and fuzzy clustering algorithms based on these metrics are developed and evaluated through Monte Carlo simulations in Sections 3 and 4, respectively. Computational efficiency is discussed in Section 5, and two interesting applications involving biological sequences are shown in Section 6. Conclusions and future work are provided in Section 7.[1]

2. Feature-based distances between categorical time series

In this section, several features providing information on the serial dependence structure of a CTS are introduced in order to define two novel distances between series.

Hereafter, $\{X_t, t \in \mathbb{Z}\}$ denotes a categorical stochastic process taking values on a number r of unordered qualitative categories, which are coded from 1 to r so that the range of the process can be seen as $\mathcal{V} = \{1, \ldots, r\}$. It is assumed that X_t is bivariate stationary, that is, the pairwise joint distribution of (X_t, X_{t-l}) is invariant in t for arbitrary l (see [4]). The marginal distribution of X_t is denoted by $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_r)^\intercal$, with $\pi_j = P(X_t = j), j = 1, \ldots, r$. Fixed $l \in \mathbb{N}$, we use the notation $p_{ij}(l) = P(X_t = i, X_{t-l} = j)$, with $i, j \in \mathcal{V}$, for the lagged bivariate probability and the notation $p_{i|j}(l) = P(X_t = i | X_{t-l} = j) = p_{ij}(l)/\pi_j$ for the lagged conditional probability.

[1] The code used to perform the analyses described throughout the paper is available in https://github.com/anloor7/PhD_degree/tree/master/r_code/paper_categorical.

2.1. Structural features for categorical processes

In order to extract suitable features characterizing the serial dependence structure of a given CTS, we first start by defining the concepts of perfect serial independence and dependence for a categorical process. Following [4], we have perfect serial independence at lag $l \in \mathbb{N}$ if and only if $p_{ij}(l) = \pi_i \pi_j$ for any $i,j \in \mathscr{V}$. On the other hand, we have perfect serial dependence at lag $l \in \mathbb{N}$ if and only if the conditional distribution $p_{\cdot|j}(l)$ is a one-point distribution for any $j \in \mathscr{V}$. Thus, in a perfect serially independent process, knowledge about X_{t-l} does not help at all in predicting the value of X_t. Conversely, in a perfect serially dependent process, the value of X_t is completely determined from X_{t-l}.

There are several association measures that describe the serial dependence structure of a categorical process at lag l. One of such measures is the so-called Cramer's v, which is defined as

$$v(l) = \sqrt{\frac{1}{r-1}\sum_{i,j=1}^{r}\frac{\left(p_{ij}(l) - \pi_i\pi_j\right)^2}{\pi_i\pi_j}}. \tag{1}$$

The quantity $v(l)$ has range $[0,1]$, with the values 0 and 1 associated with the cases of perfect serial independence and perfect serial dependence at lag l, respectively. Note that the numerator appearing in the summation of (1) measures the deviation of $p_{ij}(l)$ from the case of serial independence between i and j at lag l.

Cramer's v summarizes the serial dependence patterns of a categorical process for every pair (i,j) and $l \in \mathbb{N}$. However, this quantity is not appropriate for characterizing a given stochastic process, since different processes can exhibit the same value of $v(l)$. A better way to characterize the process X_t is by considering the matrix $\boldsymbol{V}(l) = \left(V_{ij}(l)\right)_{1\leqslant i,j\leqslant r}$, where

$$V_{ij}(l) = \frac{\left(p_{ij}(l) - \pi_i\pi_j\right)^2}{\pi_i\pi_j}. \tag{2}$$

In this way, the r^2 elements in the summation of (1) are separately considered, and a much richer picture of the underlying dependence structure of X_t is available.

The elements of the matrix $\boldsymbol{V}(l)$ give information about the so-called *unsigned* dependence of the process. However, it is often useful to know whether a process tends to stay in the state it has reached or, on the contrary, the repetition of the same state after l steps is infrequent. This motivates the concept of *signed* dependence, which arises as an analogy of the autocorrelation function of a numerical process, since such quantity can take either positive or negative values. Provided that perfect serial dependence holds, we have perfect *positive* (*negative*) serial dependence if $p_{i|i}(l) = 1$ ($p_{i|i}(l) = 0$) for all $i \in \mathscr{V}$. The reader is referred to [4] for more details about the concepts of unsigned and signed serial dependence.

Since $\boldsymbol{V}(l)$ does not shed light on the signed dependence patterns, it would be valuable to complement the information contained in $\boldsymbol{V}(l)$ with features describing signed dependence. In this regard, a common measure of signed serial dependence at lag l is the Cohen's κ, which takes the form

$$\kappa(l) = \frac{\sum_{j=1}^{r}\left(p_{jj}(l) - \pi_j^2\right)}{1 - \sum_{j=1}^{r}\pi_j^2}. \tag{3}$$

The range of $\kappa(l)$ is given by $\left[-\frac{\sum_{j=1}^{r}\pi_j^2}{1-\sum_{j=1}^{r}\pi_j^2}, 1\right]$, with the lower and upper bounds associated with the cases of perfect negative and perfect positive dependence, respectively. For instance, in the latter case, we have $\sum_{j=1}^{r} p_{jj}(l) = \sum_{j=1}^{r} p_{j|j}(l)\pi_j = \sum_{j=1}^{r}\pi_j = 1$, so $\kappa(l)$ takes the value of 1.

Proceeding as with $v(l)$, the quantity $\kappa(l)$ can be decomposed in order to obtain a more detailed representation of the signed dependence patterns of the process. In this way, we consider the vector $\boldsymbol{\mathscr{K}}(l) = (\mathscr{K}_1(l), \ldots, \mathscr{K}_r(l))$, where each $\mathscr{K}_i$, for $i = 1, \ldots, r$, is defined as

$$\mathscr{K}_i(l) = \frac{p_{ii}(l) - \pi_i^2}{1 - \sum_{j=1}^{r}\pi_j^2}. \tag{4}$$

In practice, the matrix $\boldsymbol{V}(l)$ and the vector $\boldsymbol{\mathscr{K}}(l)$ must be estimated from a T-length realization of the process, $\{X_1, \ldots, X_T\}$. To this aim, we consider estimators of π_i and $p_{ij}(l)$, $\widehat{\pi}_i$ and $\widehat{p}_{ij}(l)$, defined as

$$\widehat{\pi}_i = \frac{N_i}{T} \text{ and } \widehat{p}_{ij}(l) = \frac{N_{ij}(l)}{T-l}, \tag{5}$$

where N_i is the number of variables X_t equal to i in the realization $\{X_1, \ldots, X_T\}$, and $N_{ij}(l)$ is the number of pairs $(X_t, X_{t-l}) = (i,j)$ in the realization $\{X_1, \ldots, X_T\}$. Hence, estimates of $\boldsymbol{V}(l)$ and $\boldsymbol{\mathscr{K}}(l)$, $\widehat{\boldsymbol{V}}(l)$ and $\widehat{\boldsymbol{\mathscr{K}}}(l)$, can be obtained by plugging

in the estimates $\widehat{\pi}_i$ and $\widehat{p}_{ij}(l)$ in (2) and (4), respectively. This leads directly to estimates of $v(l)$ and $\kappa(l)$, denoted by $\widehat{v}(l)$ and $\widehat{\kappa}(l)$, whose asymptotic distributions have been studied for the i.i.d. case by [31,32]. Note that, by considering $\widehat{\boldsymbol{V}}(l)$ and $\widehat{\mathscr{K}}(l)$, a complete picture of the serial dependence patterns of a CTS is provided.

An alternative way of describing the dependence structure of the process $\{X_t, t \in \mathbb{Z}\}$ is to take into consideration its equivalent representation as a multivariate binary process. The so-called *binarization* of $\{X_t, t \in \mathbb{Z}\}$ is carried out as follows. Let $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_r \in \{0,1\}^r$ be unit vectors such that $\boldsymbol{e}_k$ has all its entries equal to zero except for a one in the k-th position, $k = 1, \ldots, r$. Then, the binary representation of $\{X_t, t \in \mathbb{Z}\}$ is given by the process $\{\boldsymbol{Y}_t = (Y_{t,1}, \ldots, Y_{t,r})^\top, t \in \mathbb{Z}\}$ such that $\boldsymbol{Y}_t = \boldsymbol{e}_j$ if $X_t = j$. Fixed $l \in \mathbb{N}$ and $i, j \in \mathscr{S}$, consider the correlation

$$\phi_{ij}(l) = Corr\big(Y_{t,i}, Y_{t-l,j}\big), \tag{6}$$

which measures linear dependence between the i-th and j-th categories with respect to the lag l. The following theorem provides some properties of the quantity $\phi_{ij}(l)$.

Theorem 1. *Let $\{X_t, t \in \mathbb{Z}\}$ be a bivariate stationary categorical process with range $\mathscr{S} = \{1, \ldots, r\}$. Then the following properties hold:*

1. *For every $i, j \in \mathscr{S}$, the function $\phi_{ij} : \mathbb{N} \to [-1, 1]$ given by $l \to \phi_{ij}(l) = Corr\big(Y_{t,i}, Y_{t-l,j}\big)$ is well-defined.*
2. $\phi_{ij}(l) = 0 \iff p_{ij}(l) = \pi_i \pi_j$.
3. $\phi_{ij}(l) = \pm 1 \iff p_{ij}(l) = \pm\sqrt{\pi_i(1-\pi_i)\pi_j(1-\pi_j)} + \pi_i\pi_j$.
4. $\phi_{ij}(l) = \sqrt{\frac{\pi_j(1-\pi_i)}{\pi_i(1-\pi_j)}} \iff p_{ij}(l) = 1$.

Proof. By construction and under stationarity, the marginal distribution of $\boldsymbol{Y}_t$ verifies

$$\boldsymbol{Y}_t \sim MULT(1; \pi_1, \ldots, \pi_r) \text{ for } \quad \text{all } t \in \mathbb{Z}. \tag{7}$$

As a result, we have

$$P(\boldsymbol{Y}_t = \boldsymbol{e}_j) = P(Y_{t,1} = 0, \ldots, Y_{t,j} = 1, \ldots, Y_{t,r} = 0) = \pi_j, \tag{8}$$

and the expectation and the covariance matrix of $\boldsymbol{Y}_t$ are given by

$$\begin{aligned} &E(\boldsymbol{Y}_t) = \sum_{i=1}^{r} \boldsymbol{e}_i \pi_i = \boldsymbol{\pi} = (\pi_1, \ldots, \pi_r)^\top, \\ &Var(\boldsymbol{Y}_t) = diag(\pi_1, \ldots, \pi_r) - \boldsymbol{\pi}\boldsymbol{\pi}^\top = \boldsymbol{\Sigma} = \big(\sigma_{ij}\big)_{1 \leqslant i,j \leqslant r}, \end{aligned} \tag{9}$$

where the $diag(\cdot)$ operator creates a square matrix whose main diagonal is the vector taken as argument and the rest of entries are equal to zero, so that

$$\sigma_{ij} = \begin{cases} \pi_i(1-\pi_i) & \text{if } i = j, \\ -\pi_i\pi_j & \text{if } i \neq j. \end{cases} \tag{10}$$

Using again the stationarity of X_t, for a lag $l \in \mathbb{N}$, we have

$$\begin{aligned} Var(\boldsymbol{Y}_t, \boldsymbol{Y}_{t-l}^\top) =& E(\boldsymbol{Y}_t \boldsymbol{Y}_{t-l}^\top) - E(\boldsymbol{Y}_t) E(\boldsymbol{Y}_{t-l}^\top) \\ =& \sum_{i=1}^{r} \sum_{j=1}^{r} \boldsymbol{e}_i \boldsymbol{e}_j^\top P(\boldsymbol{Y}_t = \boldsymbol{e}_i, \boldsymbol{Y}_{t-l} = \boldsymbol{e}_j) - \boldsymbol{\pi}\boldsymbol{\pi}^\top \\ =& \sum_{i=1}^{r} \sum_{j=1}^{r} \boldsymbol{e}_i \boldsymbol{e}_j^\top p_{ij}(l) - \boldsymbol{\pi}\boldsymbol{\pi}^\top \\ =& \boldsymbol{\Sigma}(l) = \big(\sigma_{ij}(l)\big)_{1 \leqslant i,j \leqslant r}, \end{aligned} \tag{11}$$

where

$$\sigma_{ij}(l) = p_{ij}(l) - \pi_i\pi_j, \ 1 \leqslant i, j \leqslant r. \tag{12}$$

By taking into account (10) and (12), the covariance terms $Var(\boldsymbol{Y}_t, \boldsymbol{Y}_{t-l}^\top)$ can be standardized to obtain

$$Corr(\boldsymbol{Y}_t, \boldsymbol{Y}_{t-l}^\top) = \boldsymbol{\Phi}(l) = \big(\phi_{ij}(l)\big)_{1 \leqslant i,j \leqslant r}, \tag{13}$$

with

$$\phi_{ij}(l) = \frac{\sigma_{ij}(l)}{\sigma_{ii}\sigma_{jj}} = \frac{p_{ij}(l) - \pi_i\pi_j}{\sqrt{\pi_i(1-\pi_i)\pi_j(1-\pi_j)}}, \ 1 \leqslant i,j \leqslant r. \tag{14}$$

Under stationarity, both the numerator and the denominator in (14) are well-defined so Property 1 holds. The definition of $\phi_{ij}(l)$ in (14) directly leads to the fulfillment of Properties 2 and 3. To show Property 4, assume now that $p_{i|j}(l) = 1$. As $p_{i|j}(l) = p_{ij}(l)/\pi_j$, we have $p_{ij}(l) = \pi_j$, hence

$$\phi_{ij}(l) = \frac{\pi_j - \pi_i\pi_j}{\sqrt{\pi_i(1-\pi_i)\pi_j(1-\pi_j)}} = \frac{\pi_j(1-\pi_i)}{\sqrt{\pi_i(1-\pi_i)\pi_j(1-\pi_j)}} = \sqrt{\frac{\pi_j(1-\pi_i)}{\pi_i(1-\pi_j)}}, \tag{15}$$

so Property 4 is met and the proof is completed. □

Remark 1. *About $\phi_{ij}(l)$ as a descriptive feature.* According to Theorem 1, $\phi_{ij}(l)$ provides valuable insights into both types of dependence, signed and unsigned, for the underlying process. In the case of perfect (unsigned) independence at lag l, we have that $p_{ij}(l) = \pi_i\pi_j$ for all $i,j \in \mathscr{V}$, and therefore $\phi_{ij}(l) = 0$ for all $i,j \in \mathscr{V}$, in accordance with Property 2 of Theorem 1. Under perfect positive dependence at lag l, $p_{i|i}(l) = 1$ for all $i \in \mathscr{V}$, and therefore $\phi_{ii}(l) = 1$ for all $i \in \mathscr{V}$ by following Property 4 of Theorem 1. The same property allows to conclude that $\phi_{ii}(l) = -\pi_i/(1-\pi_i)$ for all $i \in \mathscr{V}$ in the case of perfect negative dependence. In sum, $\phi_{ij}(l)$ evaluates unsigned dependence when $i \neq j$ and signed dependence when $i = j$.

Remark 2. *Relationship between $\phi_{ij}(l)$ and $V_{ij}(l)$.* From (2) and (14), it follows that $\phi_{ij}^2(l) = V_{ij}(l)/((1-\pi_i)(1-\pi_j))$. This way, $\phi_{ij}^2(l)$ is obtained by correcting $V_{ij}(l)$ by means of the marginal probabilities of categories i and j. Note that, in order to discriminate between dependence patterns, $\phi_{ij}(l)$ is indeed a more informative feature than $\phi_{ij}^2(l)$ and, in turn, than $V_{ij}(l)$ (different serial patterns for two given categories i and j leading to different $\phi_{ij}(l)$ could exhibit identical $\phi_{ij}^2(l)$).

The matrix $\mathbf{\Phi}(l)$ appearing in the proof of Theorem 1 can be directly estimated by means of $\widehat{\mathbf{\Phi}}(l) = \left(\widehat{\phi}_{ij}(l)\right)_{1\leqslant i,j\leqslant r}$, where the estimates $\widehat{\phi}_{ij}(l)$ are computed as

$$\widehat{\phi}_{ij}(l) = \frac{\widehat{p}_{ij}(l) - \widehat{\pi}_i\widehat{\pi}_j}{\sqrt{\widehat{\pi}_i(1-\widehat{\pi}_i)\widehat{\pi}_j(1-\widehat{\pi}_j)}}, \tag{16}$$

with $\widehat{\pi}_i$ and $\widehat{p}_{ij}(l)$ given in (5). It is worth remarking that the numerator in (16) is more efficiently computed by using the realization $\{\boldsymbol{Y}_1,\ldots,\boldsymbol{Y}_T\}$ of binary vectors obtained from the realization $\{X_1,\ldots,X_T\}$, that is, by computing

$$\frac{1}{T-l}\sum_{k=1}^{T-l} Y_{k+l,i}Y_{k,j} - \frac{1}{T^2}\left(\sum_{k'=1}^{T} Y_{k',i}\right)\left(\sum_{k'=1}^{T} Y_{k',j}\right). \tag{17}$$

2.2. Motivating example

In this section, we illustrate the high ability of the features introduced in the previous section to differentiate between categorical processes. To this end, let us consider two different three-state MC, denoted by Process 1 and Process 2, with transition matrices $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$, respectively, given by

$$\begin{aligned} \boldsymbol{P}_1 &= Mat^3(0.1, 0.8, 0.1, 0.6, 0.2, 0.2, 0.3, 0.4, 0.3), \\ \boldsymbol{P}_2 &= Mat^3(0.9, 0.05, 0.05, 0.05, 0.9, 0.05, 0.025, 0.025, 0.95), \end{aligned} \tag{18}$$

where the operator $Mat^k, k \in \mathbb{N}$, transforms a vector into a square matrix of order k by sequentially placing the corresponding numbers by columns.

As a first step, the stationary distributions of Processes 1 and 2 were calculated by solving the invariance equations for matrices $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$, resulting the marginal distributions $\boldsymbol{\pi}_1 = (0.3636, 0.4545, 0.1818)^\top$ and $\boldsymbol{\pi}_2 = (0.25, 0.25, 0.50)^\top$. To examine the dispersion of each distribution, we obtained the values of the Gini index, which is defined by

$$G_i = \frac{r}{r-1}\left(1 - \boldsymbol{\pi}_i^\top\boldsymbol{\pi}_i\right), \ \ i = 1, 2. \tag{19}$$

The Gini index has range $[0, 1]$, with values close to 0 indicating minimal dispersion (i.e., similarity to a one-point distribution), and values close to 1 indicating maximal dispersion (i.e., similarity to a uniform distribution). The computed values resulted $G_1 = 0.9423$ for Process 1 and $G_2 = 0.9375$ for Process 2, indicating that both processes exhibit a very high and similar amount of dispersion. Note that, in this toy example, the marginal distributions are themselves an effective tool to differentiate between the underlying Markov models.

With respect to the serial patterns, note that the dependence structure of both processes is determined by only one lag, $l = 1$. In view of the transition matrices $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$, it is clear that both Markov processes exhibit a distinct behavior. For instance, Process 2 is expected to show a substantial degree of positive serial dependence, since the three diagonal elements in the matrix $\boldsymbol{P}_2$ are close to one. On the contrary, some amount of negative dependence is anticipated for Process 1, as the diagonal elements of matrix $\boldsymbol{P}_1$ indicate low transition probabilities between the same state. In addition, compared to $\boldsymbol{P}_1$, the structure of matrix $\boldsymbol{P}_2$ indicates a larger deviation from the case of serial independence, corresponding here to a transition matrix with all entries equal to $1/3$. Therefore, features indicating departure from the independence case (e.g., $V_{ij}(1), i,j \in \{1,2,3\}$) are expected to take larger values for Process 2.

In order to illustrate the types of dependence arising in each process, values of Cramer's v and Cohen's κ were computed in both cases with respect to lags from 1 to 10. The results are shown in Fig. 1, where blue and orange colors are used for Processes 1 and 2, respectively. As expected, Cramer's v takes smaller values for Process 1, indicating that this process is closer to the case of serial independence. As for the signed dependence, Cohen's κ is able to distinguish between the positive dependence of Process 2 and the negative dependence of Process 1 when odd lags are considered.

After generating a large sample size $\left(T = 10^6\right)$ realization of each one of the processes above, estimates of the features presented in Section 2.1 were obtained. Specifically, we computed the quantities $\hat{V}_{ij}(1), \widehat{\mathscr{K}}_i(1), \hat{\phi}_{ij}(1), i,j = 1,2,3$, for both processes. The corresponding values are displayed in Fig. 2. A logarithmic scale was employed in the case of $\hat{V}_{ij}(1)$ for comparison purposes, since the values of these estimates clearly differ between Processes 1 and 2. A label was incorporated next to each bar of Fig. 2 to indicate the categories involved in the corresponding estimate. For instance, the label "13" in the top left panel implies that the associated bar refers to the quantity $\log\left(\hat{V}_{13}(1)\right)$ obtained from the realization of Process 1.

Overall, the three types of descriptive measures are markedly different for both processes. With regards to the features based on Cramer's v (first column), it is observed that the values of $\hat{V}_{ij}(1)$ are generally higher for Process 2, which is expected since, as stated previously, Process 1 is closer to a serially independent model than Process 2. The quantities $\widehat{\mathscr{K}}_i(1)$ (second column) indicate a moderate degree of negative dependence for Process 1 and a substantial amount of positive dependence for Process 2. Note that the value $\sum_{i=1}^{3} \widehat{\mathscr{K}}_i(1)$ is the estimated Cohen's κ, which is expected to take a value close to 1 in the case of perfect positive dependence. Finally, the estimates based on the binarization process (third column) summarize both previous columns. In fact, while the first three bars clearly discriminate the negative dependence of Process 1 from the positive dependence of Process 2 (see Remark 1 above), the remaining bars suggest that Process 2 exhibits a greater deviation from the serial independence case.

In summary, this toy example highlights the usefulness of the features $\hat{V}_{ij}(l), \widehat{\mathscr{K}}_i(l)$ and $\hat{\phi}_{ij}(l)$ for distinguishing between dissimilar dependence structures of categorical time series.

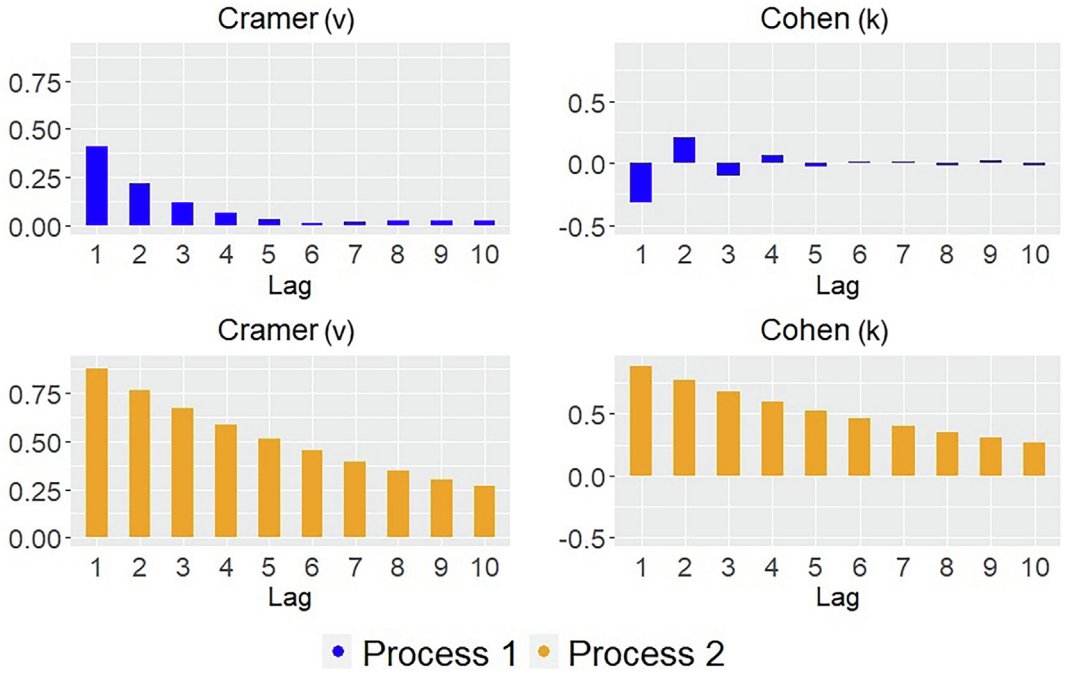

Fig. 1. Values of Cramer's $v(l)$ and Cohen's $\kappa(l)$ for Processes 1 and 2 and lags ranging from $l = 1$ to $l = 10$.

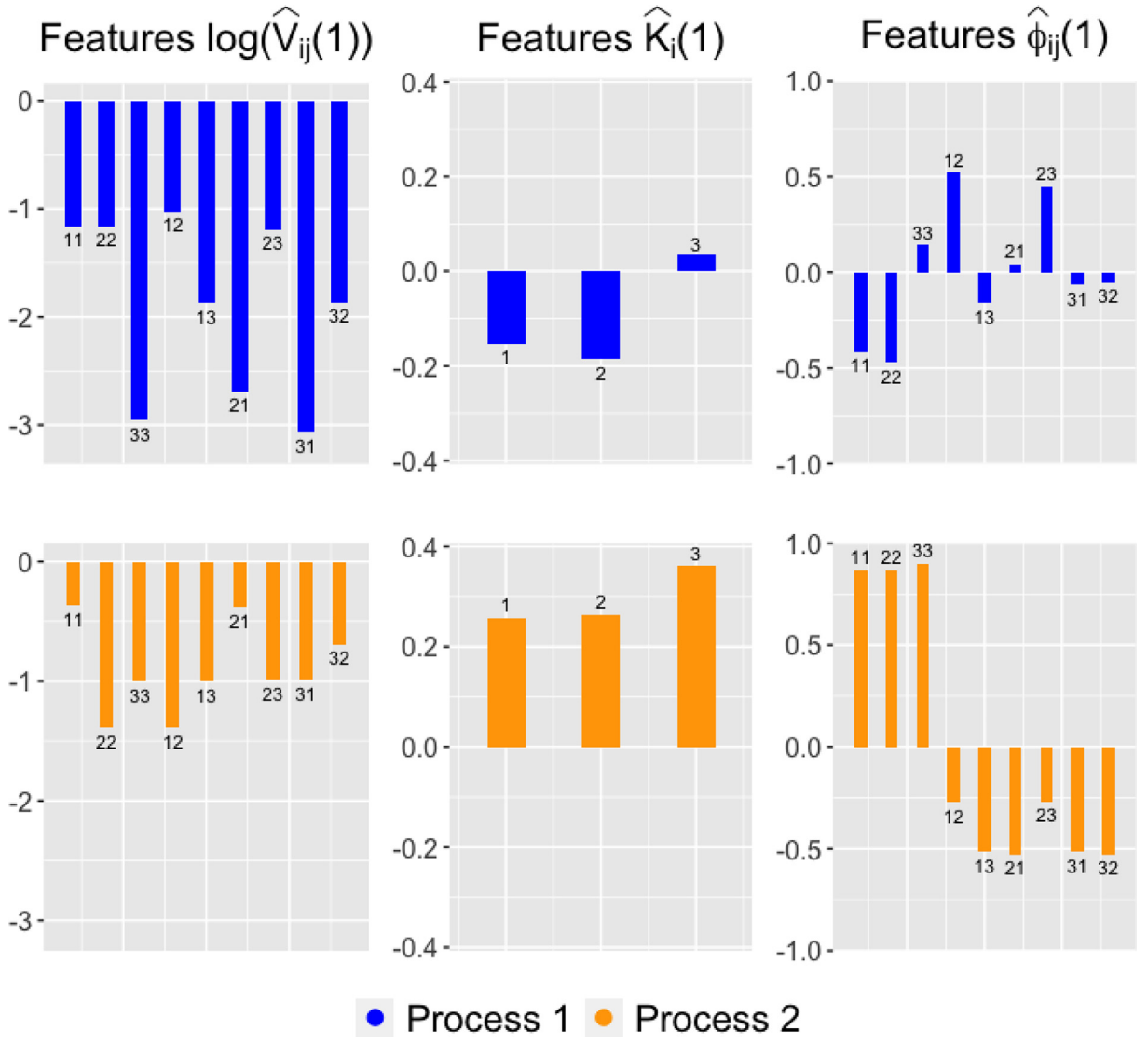

Fig. 2. Estimates of $\log(V_{ij}(1))$, $\mathscr{K}_i(1)$ and $\phi_{ij}(1)$ for large sample size realizations of Processes 1 and 2. A label was incorporated next to each bar to indicate the categories involved in the estimation.

2.3. Two innovative dissimilarities between CTS

In this section, we introduce two distance measures between categorical series based on the features described in Section 2.1 and illustrated in Section 2.2.

Suppose we have a pair of CTS, $X_t^{(1)}$ and $X_t^{(2)}$, and consider a set of L lags, $\mathscr{L} = \{l_1, \ldots, l_L\}$. A dissimilarity based on Cramer's v and Cohen's κ, so-called d_{CC}, is defined as

$$d_{CC}\left(X_t^{(1)}, X_t^{(2)}\right) = \sum_{k=1}^{L}\left[\left\| vec\left(\widehat{\boldsymbol{V}}(l_k)^{(1)} - \widehat{\boldsymbol{V}}(l_k)^{(2)}\right)\right\|^2 + \left\|\widehat{\mathscr{K}}(l_k)^{(1)} - \widehat{\mathscr{K}}(l_k)^{(2)}\right\|^2\right] + \left\|\widehat{\boldsymbol{\pi}}^{(1)} - \widehat{\boldsymbol{\pi}}^{(2)}\right\|^2, \tag{20}$$

where the superscripts (1) and (2) are used to indicate that the corresponding estimations are obtained with respect to the realizations $X_t^{(1)}$ and $X_t^{(2)}$, respectively. The metric d_{CC} combines (estimates of) the features $V_{ij}(l)$ in (2) with (estimates of) the quantities $\mathscr{K}_i(l)$ in (4). This combination often results in improved discriminative ability, since both sets of measures provide complementary information as shown in Section 2.2 (see Fig. 2).

An alternative distance measure relying on the binarization of the processes, so-called d_B, is defined as

$$d_B\left(X_t^{(1)}, X_t^{(2)}\right) = \sum_{k=1}^{L} \left\| vec\left(\widehat{\mathbf{\Phi}}(l_k)^{(1)} - \widehat{\mathbf{\Phi}}(l_k)^{(2)}\right)\right\|^2 + \|\widehat{\boldsymbol{\pi}}^{(1)} - \widehat{\boldsymbol{\pi}}^{(2)}\|^2. \tag{21}$$

The distance d_B jointly considers signed and unsigned dependence (see Remark 1 and Fig. 2), thus evaluating discrepancies between the whole serial dependence patterns of both categorical series.

Remark 3. *Independent consideration of metrics d_{CC} and d_B.* Indeed, d_{CC} and d_B could be combined to compare simultaneously the three sets of features $\hat{\phi}_{ij}(l), \hat{V}_{ij}(l)$ and $\widehat{\mathscr{K}}_i(l)$ by defining the dissimilarity $d_{COMB} = d_{CC} + d_B - \|\widehat{\boldsymbol{\pi}}^{(1)} - \widehat{\boldsymbol{\pi}}^{(2)}\|^2$. However, several numerical experiments have revealed that, in most cases, the clustering accuracy using the combined distance d_{COMB} is lower than the one achieved with the algorithms based on one of the individual distances, d_{CC} or d_B. This is due to the fact that redundant information is supplied when all features are jointly used. In fact, the serial dependence patterns captured by $\phi_{ij}(l)$ are also explained by either $V_{ij}(l)$ or $\mathscr{K}_i(l)$, and conversely. Since the use of redundant features is known to be counterproductive in clustering and classification contexts, it can be concluded that the independent consideration of metrics d_{CC} and d_B is a more suitable approach.

Remark 4. *Consideration of the marginal probabilities.* Note that a term measuring discrepancies between the marginal distributions appears in the definition of both metrics, d_{CC} and d_B. In fact, this term can play an important role to measure dissimilarity. Assume that X_t and Y_t are two bivariate stationary categorical processes such that, for all $l \in \mathbb{N}$, (X_t, X_{t-l}) has the same distribution than (Y_t, Y_{t-l}), which is given by the joint probabilities $p_{ij}(l)$. Then, since the probabilities $P(X_t = i)$ and $P(Y_t = i)$ can be expressed as $\pi_i = \sum_{j=1}^{r} p_{ij}(l)$ for all $i = 1, \ldots, r$ and any lag l, we conclude that X_t and Y_t have the same marginal distribution. Therefore, taking into account the marginal probabilities does not pervert the distances d_{CC} and d_B in the case of equal processes. On the other hand, it is possible that two processes with different lagged bivariate probabilities can be distinguished through d_{CC} and d_B only by virtue of the marginal probabilities. For instance, consider simply two categorical process formed by i.i.d. elements and having different marginal distributions. In such a case, $V_{ij}(l) = \mathscr{K}_i(l) = \phi_{ij}(l) = 0$ for both processes, and both distances d_{CC} and d_B will draw out different values for realizations of these processes due to the terms involving the marginal probabilities.

Remark 5. *Advantages of feature-based distances.* Both proposed metrics rely on two steps: (i) each time series is replaced by a set of extracted features and (ii) a standard distance between both sets of features is computed. This type of metrics, usually referred to as feature-based dissimilarities, have several advantages including dimensionality reduction, low computational complexity, selection of the most suitable features for a given context and possibility of comparing series with different lengths, since the computation of the distance takes place in the reduced space. It is worth remarking that this is not the case with many other distance measures for categorical series, for instance, metrics based on raw data, which usually involve high computational cost and require both series to have the same length.

For a given set of categorical series, the distances d_{CC} and d_B can be used as input for traditional clustering algorithms. This way, procedures for grouping a set of CTS according to the underlying dependence structures can be developed.

3. Partitioning around medoids clustering of categorical time series

In the following, the behavior of d_{CC} and d_B in hard clustering is examined through a comprehensive simulation study. After describing in detail the simulation mechanism and the assessment criteria, the main results are reported and properly discussed. A discussion regarding the selection of the set $\mathscr{L}$ is also provided. Finally, the performance of the metrics is evaluated in scenarios with a greater degree of complexity.

3.1. Experimental design

We wish to perform clustering on a set of s categorical times series, $\mathscr{S} = \left\{X_t^{(1)}, \ldots, X_t^{(s)}\right\}$, supposing that the target is to group together series with the same underlying process. Therefore, the clustering task is determined by the dynamic behaviors of the CTS. We assume the existence of C clusters in the collection $\mathscr{S}$, denoted by $\mathscr{C} = \{\mathscr{C}_1, \ldots, \mathscr{C}_C\}$. The most popular partitioning-based procedure is likely the k-means algorithm. However, this approach is not a suitable choice in our context because the average vectors of the estimated features involved in d_{CC} and d_B do not necessarily characterize a categorical process. Moreover, it is often interesting to find prototype objects for each cluster, i.e., time series summarizing the different dynamic patterns. A usual way to address these issues is to consider a k-medoids-based procedure, in which the representative elements must belong to the original set of CTS. Based on previous considerations, we have examined the behavior of the proposed metrics by using the classical version of the well-known PAM algorithm [33]. A sketch of the corresponding clustering procedure is shown in Algorithm 1.

Algorithm 1: The PAM algorithm based on the proposed distances.

1: Fix C and $d^* \in \{d_{CC}, d_B\}$
2: Pick the initial medoids $\widetilde{\mathscr{S}} = \left\{\widetilde{X}_t^{(1)}, \ldots, \widetilde{X}_t^{(C)}\right\}$ and define the initial clustering partition $\mathscr{C} = \{\mathscr{C}_1, \ldots, \mathscr{C}_C\}$
3: Compute the value of the objective function as $J = \sum_{c=1}^{C} \sum^{s}_{\substack{i=1: \\ X_t^{(i)} \in \mathscr{C}_c}} d^*\left(X_t^{(i)}, \widetilde{X}_t^{(c)}\right)$
4: Define $J^* = matrix(Inf, nrow = C, ncol = s - C)$
5: **repeat**
6: Set $J_{OLD} = J$ {Store the current cost}
7: **for** j = 1 **to** C **do**
8: **for** k = 1 **to** $s : X_t^{(k)} \notin \widetilde{\mathscr{S}}$ **do**
9: Replace the medoid $\widetilde{X}_t^{(j)}$ by the series $X_t^{(k)}$
10: Update $\widetilde{\mathscr{S}}$ and $\mathscr{C}$ (in auxiliary variables)
11: $J^*[j,k] = \sum_{c=1}^{C} \sum^{s}_{\substack{i=1: \\ X_t^{(i)} \in \mathscr{C}_c}} d^*\left(X_t^{(i)}, \widetilde{X}_t^{(c)}\right)$
12: **end for**
13: **end for**
14: $(j^*, k^*) = \text{argmin}_{(j,k)} J^*[j,k]$
15: $J = J^*[j^*, k^*]$ {Update the cost}
16: Replace the medoid $\widetilde{X}_t^{(j^*)}$ by the series $X_t^{(k^*)}$
17: Update $\widetilde{\mathscr{S}}$ and $\mathscr{C}$
18: **until** $J_{OLD} \leqslant J$
19: **return** The resulting clustering partition

Several simulations were conducted to assess the performance of both dissimilarities. The simulated scenarios encompass a broad variety of generating processes. In particular, three setups were considered, namely clustering of: (i) MC, (ii) HMM, and (iii) New Discrete ARMA (NDARMA) processes. The choice of such type of processes was made with the goal of performing the evaluation task in a fair and general manner. Indeed, the three selected settings are essential in several fields (specific applications of the three types of processes can be seen in [7], Chapter 7). The specific generating models for each class of processes are given below.

Scenario 1. Clustering of MC. Consider four three-state MC, so-called MC_1, MC_2, MC_3 and MC_4, with respective transition matrices $\boldsymbol{P}_1^1, \boldsymbol{P}_2^1, \boldsymbol{P}_3^1$ and $\boldsymbol{P}_4^1$ given by

$$\begin{aligned}
\boldsymbol{P}_1^1 &= Mat^3(0.1, 0.8, 0.1, 0.5, 0.4, 0.1, 0.6, 0.2, 0.2),\\
\boldsymbol{P}_2^1 &= Mat^3(0.1, 0.8, 0.1, 0.6, 0.3, 0.1, 0.6, 0.2, 0.2),\\
\boldsymbol{P}_3^1 &= Mat^3(0.05, 0.90, 0.05, 0.05, 0.05, 0.90, 0.90, 0.05, 0.05),\\
\boldsymbol{P}_4^1 &= Mat^3(1/3, 1/3, 1/3, 1/3, 1/3, 1/3, 1/3, 1/3, 1/3).
\end{aligned} \tag{22}$$

Scenario 2. Clustering of HMM. Consider the bivariate process $(X_t, Q_t)_{t\in\mathbb{Z}}$, where Q_t stands for the hidden states and X_t for the observable random variables. Process $(Q_t)_{t\in\mathbb{Z}}$ constitutes an homogeneous MC. Both $(X_t)_{t\in\mathbb{Z}}$ and $(Q_t)_{t\in\mathbb{Z}}$ are assumed to be count processes with range $\{1, \ldots, r\}$. Process $(X_t, Q_t)_{t\in\mathbb{Z}}$ is supposed to verify the three classical assumptions of a HMM (see, e.g., Section 7.3 in [7]). Based on previous considerations, let HMM_1, HMM_2, HMM_3 and HMM_4 be four three-state HMM with respective transition matrices $\boldsymbol{P}_1^2, \boldsymbol{P}_2^2, \boldsymbol{P}_3^2$ and $\boldsymbol{P}_4^2$ and emission matrices $\boldsymbol{E}_1^2, \boldsymbol{E}_2^2, \boldsymbol{E}_3^2$ and $\boldsymbol{E}_4^2$ given by

$$\begin{aligned}
\boldsymbol{P}_1^2 &= Mat^3(0.05, 0.90, 0.05, 0.05, 0.05, 0.90, 0.90, 0.05, 0.05),\\
\boldsymbol{P}_2^2 &= Mat^3(0.05, 0.90, 0.05, 0.05, 0.05, 0.90, 0.90, 0.05, 0.05),\\
\boldsymbol{P}_3^2 &= Mat^3(0.1, 0.7, 0.2, 0.4, 0.4, 0.2, 0.4, 0.3, 0.3),\\
\boldsymbol{P}_4^2 &= Mat^3(1/3, 1/3, 1/3, 1/3, 1/3, 1/3, 1/3, 1/3, 1/3),\\
\boldsymbol{E}_1^2 &= Mat^3(0.05, 0.90, 0.05, 0.05, 0.05, 0.90, 0.90, 0.05, 0.05),\\
\boldsymbol{E}_2^2 &= Mat^3(0.1, 0.8, 0.1, 0.5, 0.4, 0.1, 0.6, 0.2, 0.2),\\
\boldsymbol{E}_3^2 &= Mat^3(0.1, 0.8, 0.1, 0.5, 0.4, 0.1, 0.6, 0.2, 0.2),\\
\boldsymbol{E}_4^2 &= Mat^3(1/3, 1/3, 1/3, 1/3, 1/3, 1/3, 1/3, 1/3, 1/3).
\end{aligned} \tag{23}$$

Scenario 3. Clustering of NDARMA processes. Let $(X_t)_{t\in\mathbb{Z}}$ and $(\epsilon_t)_{t\in\mathbb{Z}}$ be two count processes with range $\{1,\ldots,r\}$ and following the equation

$$X_t = \alpha_{t,1}X_{t-1} + \ldots + \alpha_{t,p}X_{t-p} + \beta_{t,0}\epsilon_t + \ldots + \beta_{t,q}\epsilon_{t-q}, \tag{24}$$

where $(\epsilon_t)_{t\in\mathbb{Z}}$ is i.i.d. with $P(\epsilon_t = i) = \pi_i$, independent of $(X_s)_{s<t}$, and the i.i.d. multinomial random vectors

$$(\alpha_{t,1},\ldots,\alpha_{t,p},\beta_{t,0},\ldots,\beta_{t,q}) \sim \text{MULT}\Big(1;\phi_1,\ldots,\phi_p,\varphi_0,\ldots,\varphi_q\Big), \tag{25}$$

are independent of $(\epsilon_t)_{t\in\mathbb{Z}}$ and of $(X_s)_{s<t}$. The considered models are 3 three-state NDARMA(2,0) processes and one three-state NDARMA(1,0) process with marginal distribution $\boldsymbol{\pi}^3 = (2/3, 1/6, 1/6)$, and corresponding probabilities in the multinomial distribution given by

$$\begin{aligned}
(\phi_1,\phi_2,\varphi_0)_1^3 &= (0.7,0.2,0.1),\\
(\phi_1,\phi_2,\varphi_0)_2^3 &= (0.1,0.45,0.45),\\
(\phi_1,\phi_2,\varphi_0)_3^3 &= (0.5,0.25,0.25),\\
(\phi_1,\varphi_0)_4^3 &= (0.2,0.8).
\end{aligned} \tag{26}$$

The simulation study was carried out as follows. For each scenario, 5 CTS of length $T \in \{200, 600\}$ were generated from each process in order to execute the clustering algorithms twice, thus allowing to analyze the impact of the series length. The resulting clustering solution produced by each considered algorithm was stored. The simulation procedure was repeated 500 times for each scenario and value of T. The computation of d_{CC} and d_B was carried out by considering $\mathscr{L} = \{1\}$ in Scenarios 1 and 2 and $\mathscr{L} = \{1,2\}$ in Scenario 3. This way, we adapted the distances to the maximum number of defining lags for the majority of clusters.

3.2. Alternative metrics and assessment criteria

To get insights into the performance of both metrics d_{CC} and d_B, we also obtained partitions by using the alternative techniques for clustering of categorical series described below.

- *Model-based approach using maximum likelihood estimation (MLE).* Let $\boldsymbol{\theta}$ be the parameter vector of a MC (Scenario 1), a HMM (Scenario 2) or an NDARMA model (Scenario 3). Each CTS $X_t^{(i)}, i = 1,\ldots,s$, is described by means of the maximum likelihood estimate of its corresponding parameter vector, $\widehat{\boldsymbol{\theta}}^{(i)}$. The distance between two CTS $X_t^{(j)}$ and $X_t^{(k)}$ is defined as the squared Euclidean distance between the vectors $\widehat{\boldsymbol{\theta}}^{(j)}$ and $\widehat{\boldsymbol{\theta}}^{(k)}$, $\|\widehat{\boldsymbol{\theta}}^{(j)} - \widehat{\boldsymbol{\theta}}^{(k)}\|^2$. We denote this dissimilarity by d_{MLE}. The distance matrix constructed using this metric is used as input to the PAM algorithm.
- *Model-based approach using mixtures.* [8] propose to cluster a set of CTS by learning a mixture of first order Markov models via the EM algorithm. The only hyperparameter of the method is the number of components, which can be identified with the number of clusters. Although this approach does not use directly a distance metric, for the sake of homogeneity, we denote the technique by d_{CZ}.
- *A clustering procedure based on DHM.* [15] introduces the DHM, which is a variant of the so-called temporal patterns optimal matching (OM) for categorical sequences. OM employs a combination of both indel (insertion and deletion) and substitution costs in order to uncover different socio-temporal patterns. Specifically, DHM uses only substitution operations with time-dependent costs inversely proportional to transition frequencies. The PAM algorithm is executed by considering the dissimilarity matrix associated with DHM. The corresponding metric is denoted by d_{DHM}.
- *A clustering technique relying on OM.* [16] constructs a modified version of the OM algorithm, so-called OMv, which weights OM's elementary operations inversely with episode length. The modified procedure substantially differs from OM when there is high variability in spell length. The OMv-based distance matrix is used to feed the PAM algorithm. The corresponding dissimilarity is denoted by d_{OMV}.
- *An hybrid framework for clustering CTS.* [19] presents a dissimilarity between categorical series which evaluates both closeness between raw categorical values and proximity between dynamic patterns. To this aim, the distance introduced by [34] is combined with a correlation-based metric between categorical sequences. An hyperparameter $k \in \mathbb{N}$ regulates the weight of each dissimilarity in the resulting measure. This way, the user can give different importances to the geometric and dynamic parts. The PAM algorithm is executed by considering the combined distance, which is denoted by d_{MV}. To apply this methodology in the simulations, we run the clustering algorithm for several values of k and selected the most accurate partition according to the Adjusted Rand Index (ARI) [35].

Note that the approach based on the distance d_{MLE} can be seen as a strict benchmark in the evaluation task. Indeed, this procedure assumes the true parametric models in each scenario when computing the parameter vector estimates, which constitutes a substantial advantage over the remaining techniques. In this regard, we are comparing the performance of both proposed dissimilarities d_{CC} and d_B with that of one of the hardest competitors in the context of CTS clustering.

The effectiveness of the clustering approaches was assessed by comparing the clustering solutions produced by the algorithms with the true clustering partition, so-called ground truth. The latter consisted of $C = 4$ clusters in all scenarios, each group including the five CTS generated from the same process. The value $C = 4$ was provided as input parameter to the PAM algorithm in the case of $d_{CC}, d_B, d_{MLE}, d_{DHM}, d_{OMV}$ and d_{MV}. As for the approach d_{CZ}, a number of 4 components were considered for the mixture model. Experimental and true partitions were compared by using three well-known external clustering quality indices, the ARI, the Jaccard Index (JI) and the Fowlkes-Mallows index (FMI) [36]. ARI index takes values in $[-1, 1]$, whereas the remaining indices are bounded between 0 and 1. In all cases, the closer to one the index, the better the quality of the clustering partition.

3.3. Results and discussion

The average values of the quality indices by taking into account the 500 simulation trials are given in Tables 3–5 for Scenarios 1, 2 and 3, respectively.

It is clear from Tables 3–5 that the dissimilarities d_{MV} and d_{DHM} lead to results substantially worse than the rest of the methods in all scenarios, suggesting that these metrics are not appropriate to differentiate between stationary categorical processes. The distance d_{OMV} attains better results than d_{MV} and d_{DHM} but it is still far from the best-performing metrics, specially when $T = 600$. Although its results for $T = 200$ are acceptable, this dissimilarity shows little improvement when increasing the series length.

The results in Table 3 indicate that d_{CC} is the most effective dissimilarity when dealing with MC, outperforming the benchmark metric d_{MLE}. Although to a lesser extent, d_B is also superior to d_{MLE} in Scenario 1, while d_{CZ} exhibits an outstanding performance when $T = 600$, with similar scores as d_{CC}.

Table 4 shows a completely different picture, with d_{CC} and d_B exhibiting a significantly better effectiveness than the rest of the metrics. The latter outperforms the former by a moderate degree, achieving virtually perfect results for the largest value of T. The distance based on estimated model coefficients, d_{MLE}, shows a poor performance, which is likely due to the fact that the estimation process for HMM suffers from some drawbacks as slow convergence, low accuracy of parameter estimation and high dependency on initial guesses [37]. As expected, d_{CZ} is affected in this scenario by model misspecification, achieving significantly worse scores than in Scenario 1.

As for Scenario 3, the results in Table 5 reveal that the model-based distance d_{MLE} attains the highest scores when $T = 200$, but it is defeated by d_B when $T = 600$. The metric d_{CZ} suffers again from model misspecification.

3.4. Analysing clustering effectiveness with respect to the set $\mathscr{L}$

Note that, in practice, the clustering algorithms based on d_{CC} and d_B require fixing the set of lags $\mathscr{L}$ involved in the computation of both dissimilarities. The results in Tables 3–5 were obtained using a specific collection of lags in each scenario. However, it is interesting to assess how the clustering accuracy fluctuates when different sets are considered. In this regard, a sensitivity analysis was performed by considering the scenarios in Section 3.1 and 5 different collections of lags, namely $\mathscr{L}_1, \ldots, \mathscr{L}_5$, with $\mathscr{L}_i = \{1, 2, \ldots, i\}$ for $i = 1, \ldots, 5$. Tables 6 and 7 contain the results for dissimilarities d_{CC} and d_B, respectively. For the sake of simplicity, only the average scores in terms of ARI index are presented.

According to the quantities in Tables 6 and 7, both metrics are quite robust with respect to the choice of $\mathscr{L}$, but they display a different behavior in each of the considered settings. In Scenario 1, the dissimilarities achieve the best results when only the first lag is selected ($\mathscr{L}_1$), and then slightly decrease their performance when more lags (noise) are incorporated in the set. A similar situation occurs in Scenario 3 when short series are generated ($T = 200$), with average scores moderately decreasing when more than two lags ($\mathscr{L}_2$) are considered. This decline is no longer observed when increasing the series length ($T = 600$), since the features involved in the computation of the distances are estimated with high accuracy. Finally, in Scenario 2, clustering accuracy moderately improves when the third lag is included ($\mathscr{L}_3$), which is due to the fact that the second process in this scenario exhibits the strongest degree of serial dependence at $l = 3$ (see Section 7.3 in [7] for a description of the serial dependence structure of a HMM). Notice that, in general, moderate deviations from the nominal lag order have very low impact on the clustering accuracy. Thus, while the optimal lag selection is a critical issue in modeling and forecasting problems, the clustering approaches based on d_{CC} and d_B exhibit a reasonable robustness to a nonoptimal choice of $\mathscr{L}$. This is a particularly nice property in our setting because the proposed clustering algorithms are model-free and no single lag selection procedure has been proven to perform well with all time series models.

In view of the mentioned robustness, the required set of lags $\mathscr{L}$ can be determined by a simple and automatic criterion, mainly satisfying two properties: applicability without prior assumptions about the generating models and computational efficiency. Note that, in the case of an arbitrary i.i.d. process, the distribution of $T(r-1)\widehat{v}^2(l)$ is approximated by a $\chi^2_{(r-1)^2}$ [31]. Hence, for a given series and a significance level α, the null hypothesis of serial independence at lag l is rejected if

$$\widehat{v}(l) > \sqrt{\frac{1}{T(r-1)}\chi^2_{(r-1)^2,1-\alpha}}, \tag{27}$$

Table 3
Scenario 1: Average values over 500 trials of 3 clustering quality indices based on several dissimilarities. For each index and value of T, the best result is shown in bold.

	$T = 200$			$T = 600$		
Method	ARI	JI	FMI	ARI	JI	FMI
d_{CC}	**0.761**	**0.697**	**0.817**	**0.917**	**0.888**	**0.936**
d_B	0.729	0.661	0.792	0.861	0.878	0.893
d_{MLE}	0.704	0.633	0.772	0.841	0.792	0.876
d_{CZ}	0.712	0.648	0.786	0.915	0.886	0.934
d_{MV}	0.406	0.363	0.665	0.379	0.363	0.650
d_{DHM}	0.169	0.243	0.401	0.145	0.245	0.424
d_{OMV}	0.581	0.524	0.702	0.599	0.539	0.715

Table 4
Scenario 2: Average values over 500 trials of 3 clustering quality indices based on several dissimilarities. For each index and value of T, the best result is shown in bold.

	$T = 200$			$T = 600$		
Method	ARI	JI	FMI	ARI	JI	FMI
d_{CC}	0.705	0.638	0.777	0.853	0.808	0.887
d_B	**0.760**	**0.701**	**0.812**	**0.963**	**0.949**	**0.971**
d_{MLE}	0.354	0.342	0.512	0.299	0.310	0.478
d_{CZ}	0.645	0.577	0.739	0.703	0.638	0.779
d_{MV}	0.089	0.175	0.323	0.062	0.175	0.301
d_{DHM}	0.142	0.213	0.352	0.180	0.248	0.406
d_{OMV}	0.540	0.489	0.667	0.581	0.524	0.702

Table 5
Scenario 3: Average values over 500 trials of 3 clustering quality indices based on several dissimilarities. For each index and value of T, the best result is shown in bold.

	$T = 200$			$T = 600$		
Method	ARI	JI	FMI	ARI	JI	FMI
d_{CC}	0.621	0.558	0.723	0.869	0.846	0.913
d_B	0.680	0.612	0.754	**0.925**	**0.901**	**0.941**
d_{MLE}	**0.727**	**0.656**	**0.788**	0.872	0.828	0.900
d_{CZ}	0.586	0.562	0.693	0.647	0.577	0.738
d_{MV}	0.035	0.167	0.292	−0.028	0.138	0.251
d_{DHM}	0.329	0.319	0.481	0.386	0.359	0.524
d_{OMV}	0.529	0.489	0.668	0.589	0.532	0.710

Table 6
Average values over 500 trials of ARI index based on d_{CC} for different sets of lags ($\mathscr{L}_j, j = 1, \ldots, 5$). *Si* denotes Scenario $i, i = 1, 2, 3$, and the theoretical set of lags is given in brackets. For each scenario and value of T, the best result is shown in bold.

	$T = 200$			$T = 600$		
Set	S1 ($\mathscr{L}_1$)	S2 ($\mathscr{L}_1$)	S3 ($\mathscr{L}_2$)	S1 ($\mathscr{L}_1$)	S2 ($\mathscr{L}_1$)	S3 ($\mathscr{L}_2$)
$\mathscr{L}_1$	**0.761**	0.705	0.621	**0.917**	0.853	0.869
$\mathscr{L}_2$	0.750	0.690	**0.677**	0.900	0.843	0.960
$\mathscr{L}_3$	0.718	0.738	0.652	0.869	**0.959**	0.969
$\mathscr{L}_4$	0.689	**0.742**	0.630	0.844	0.953	**0.973**
$\mathscr{L}_5$	0.674	0.729	0.604	0.823	0.953	0.969

where $\chi^2_{g,1-\alpha}$ denotes the $(1-\alpha)$-quantile of the distribution χ^2_g. This way, a simple criterion for selecting $\mathscr{L}$ consists of determining the significant lags for each CTS in the dataset according to (27), and then fixing a maximum lag for all of them. Specifically, given the set $\mathscr{S} = \left\{X_t^{(1)}, \ldots, X_t^{(s)}\right\}$ of CTS subject to clustering, we propose to select $\mathscr{L}$ as follows.

1. Fix $\alpha > 0$ and a maximum lag $L_{\text{Max}} \in \mathbb{N}$. Using the Bonferroni's adjustment for multiple comparisons, compute the corrected significance level $\alpha\prime = \alpha/(sL_{\text{Max}})$.
2. For each series $X_t^{(i)} \in \mathscr{S}$:

Table 7
Average values over 500 trials of ARI index based on d_B for different sets of lags ($\mathscr{L}_j, j = 1, \ldots, 5$). *Si* denotes Scenario $i, i = 1, 2, 3$, and the theoretical set of lags is given in brackets. For each scenario and value of T, the best result is shown in bold.

	$T = 200$			$T = 600$		
Set	S1 ($\mathscr{L}_1$)	S2 ($\mathscr{L}_1$)	S3 ($\mathscr{L}_2$)	S1 ($\mathscr{L}_1$)	S2 ($\mathscr{L}_1$)	S3 ($\mathscr{L}_2$)
$\mathscr{L}_1$	**0.729**	0.760	0.680	**0.861**	0.963	0.925
$\mathscr{L}_2$	0.709	0.796	**0.682**	0.851	0.969	0.928
$\mathscr{L}_3$	0.698	**0.837**	0.667	0.826	**0.972**	0.944
$\mathscr{L}_4$	0.689	0.815	0.652	0.796	0.962	**0.951**
$\mathscr{L}_5$	0.670	0.784	0.627	0.774	0.956	0.948

2.1. Determine the set $\mathscr{L}_{(i)}$ formed for all the lags $l_i \in \{1, 2, \ldots, L_{\text{Max}}\}$ for which the decision rule (27) leads to rejection at level $\alpha/$.
2.2. Select the lag $L_i \in \mathscr{L}_{(i)}$ maximizing the estimated Cramer's v, i.e., $\hat{v}(L_i) = \max\{\hat{v}(l_i), l_i \in \mathscr{L}_{(i)}\}$.
3. Set $L^* = \max\{L_1, \ldots, L_s\}$ and $\mathscr{L} = \{1, 2, \ldots, L^*\}$.

Some remarks concerning the previous procedure are given below. The Bonferroni correction is considered in Step 1 to address the problem of multiple comparisons, since sL_{Max} statistical tests are simultaneously performed. The use of a conservative rule is motivated because frequently a few lags are sufficient to characterize the serial dependence. Nonetheless, other procedures ensuring that the family-wise error rate is at most α could be employed. In Step 2.2, the lag maximizing the estimated Cramer's v is selected to avoid including redundant features, since dependence at lower lags usually produces (less strong) dependence at higher lags (e.g., the serial dependence structure of a MC). Lastly, in Step 3, L^* is the maximum lag within $\mathscr{L}$. By construction, L^* is necessarily a significant lag for one or several series, although indeed some series might not exhibit significant serial dependence at L^* or lower lags. However, this is not an issue because the corresponding estimated features are expected to be close to zero for these series.

We extended the simulation study to gain insights into the behavior of the previous strategy. Considering Scenarios 1, 2 and 3 again, the procedure was run by setting $\alpha = 0.05$ and $L_{\text{Max}} = 5$. According to 500 simulation trials for each value of T, the proportion of times that each set of lags was selected is provided in Table 8. The results are consistent with the ones in Tables 6 and 7, which indicates that the proposed method frequently succeeds in determining the optimal set of lags. In fact, the sets $\mathscr{L}_1$ and $\mathscr{L}_2$ were selected almost 100% of the trials in Scenarios 1 and 3, respectively. A different situation is observed in Scenario 2, where the series length greatly influences the choice of $\mathscr{L}$. Thus, $\mathscr{L} = \{1\}$ was often selected when $T = 200$ because the significant dependence at lag $l = 3$ exhibited by the series in the second cluster was not generally detected. This dependence was more easily identified when increasing the series length, which accounts for the selection of $\mathscr{L}_3$ most of the times when $T = 600$. Notice that the ability to detect the dependence at the third lag could be improved by considering a less conservative multiple testing correction. Similar conclusions were obtained by using alternative values of α (e.g., $\alpha = 0.01$ or $\alpha = 0.10$).

3.5. Additional experiments. Scenarios with a higher degree of complexity

To obtain a more comprehensive evaluation of the proposed clustering methods, more challenging setups were constructed by increasing the complexity of Scenarios 1, 2 and 3. First, a new scenario with 6 clusters defined by models MC_1, MC_2 and MC_3 in Scenario 1 and models HMM_1, HMM_2 and HMM_3 in Scenario 2 was considered. This way, the new setup involves two different types of generating processes (MC and HMM), which makes the clustering task substantially harder. Notice that we decided to combine MC and HMM because the respective vectors of estimated model coefficients are easily comparable (HMM can be seen as an extension of MC), and thus introducing a common d_{MLE} metric for the new scenario is quite straightforward.

The experiments were carried out as in previous sections but varying some parameters. Ten series of length $T \in \{350, 700\}$ were simulated from each generating process and the number of clusters was set to $C = 6$. The metrics and performance indices used in Section 3.3 were also considered here except for d_{MV}, which was removed from the analysis due to its poor behavior and high computational cost. As the MLE-based vectors for MC and HMM differ in length (lengths 9 and 18, respectively), the vectors in the former case were properly padded with zeros in order to compute d_{MLE}. This way, the estimated transition probabilities of both processes can be compared and the emission probabilities are set to zero in MC (which is reasonable since they only exist in HMM). Metrics d_{CC} and d_B were computed by considering $\mathscr{L} = \{1\}$ and the simulation experiment was run 500 times.

Table 9 shows the average values of the clustering quality indices for the new scenario. Metrics d_{CC} and d_B outperform the remaining ones by a large degree, while no significant differences between both of them are observed. Metrics d_{MLE} and d_{CZ} display acceptable scores, but they are negatively affected by the complexity of the current scenario. Lastly, d_{DHM} and d_{OMV} exhibit a very poor behavior in all cases. Interestingly, the proposed distances are the ones showing the greatest

Table 8
Proportion of times that each set $\mathscr{L}_i$ was selected according to the proposed criterion. For each scenario and value of T, the largest rate is shown in bold. A significance level $\alpha = 0.05$ was considered.

		$\mathscr{L}_1$	$\mathscr{L}_2$	$\mathscr{L}_3$	$\mathscr{L}_4$	$\mathscr{L}_5$
Scenario 1	$T = 200$	**0.988**	0	0.006	0.002	0.004
	$T = 600$	**0.980**	0.004	0.004	0.004	0.008
Scenario 2	$T = 200$	**0.750**	0.044	0.172	0.018	0.016
	$T = 600$	0.070	0.026	**0.816**	0.058	0.030
Scenario 3	$T = 200$	0	**0.996**	0.002	0	0.002
	$T = 600$	0	**0.998**	0	0	0.002

Table 9
Complex scenario ($C = 6$): Average values over 500 trials of 3 clustering quality indices based on several dissimilarities. For each index and value of T, the best result is shown in bold.

	$T = 350$			$T = 700$		
Method	ARI	JI	FMI	ARI	JI	FMI
d_{CC}	**0.718**	**0.621**	**0.766**	0.799	0.715	0.833
d_B	0.701	0.601	0.750	**0.803**	**0.720**	**0.835**
d_{MLE}	0.598	0.501	0.675	0.613	0.516	0.687
d_{CZ}	0.629	0.540	0.708	0.687	0.598	0.753
d_{DHM}	0.101	0.154	0.274	0.095	0.156	0.281
d_{OMV}	0.375	0.323	0.497	0.412	0.353	0.534

Table 10
Randomized scenario ($C = 5$): Average values over 500 trials of 3 clustering quality indices based on several dissimilarities. For each index and value of T, the best result is shown in bold.

	$T = 350$			$T = 700$		
Method	ARI	JI	FMI	ARI	JI	FMI
d_{CC}	**0.759**	**0.685**	**0.807**	0.829	0.769	0.863
d_B	0.745	0.667	0.795	**0.833**	**0.771**	**0.865**
d_{MLE}	0.627	0.547	0.711	0.652	0.572	0.730
d_{CZ}	0.661	0.591	0.745	0.694	0.629	0.773
d_{DHM}	0.113	0.183	0.315	0.105	0.184	0.322
d_{OMV}	0.408	0.365	0.541	0.445	0.396	0.577

improvement when increasing the series length. These results corroborate the great performance of d_{CC} and d_B in challenging scenarios.

A second additional scenario with $C = 5$ clusters was constructed to examine the effect of varying the number of groups. To this aim, we considered the simulation scheme previously described and introduced some degree of uncertainty. Specifically, at each trial, we removed the ten series associated with one particular generating process, which was chosen at random. Note that this mechanism creates a challenging setup where the whole underlying structures in the CTS dataset are successively different. This way, robustness of the clustering techniques against slight changes in the dependence patterns of the database is being evaluated. Results for this randomized scenario are displayed in Table 10.

Average scores in Table 10 ($C = 5$) are very similar to the ones in Table 9 ($C = 6$) but there is a slight improvement in the performance of all dissimilarities. This is reasonable, since, usually, the lower the number of clusters, the simpler for the clustering algorithms to identify the true partition. In brief, these results illustrate the robustness of the proposed distances against the elimination of a certain generating process from the database.

To summarize, the numerical experiments carried out throughout this section clearly show the high ability of the proposed measures, d_{CC} and d_B, to discriminate between a broad variety of categorical processes. In fact, the PAM algorithm based on both distances exhibits an excellent clustering accuracy under a wide variety of settings, including different values for the number of clusters, series per cluster and series length, among others. Compared to the model-based procedures, which take advantage of knowing the true underlying models, the proposed methods either produce better results or show a similar behavior. Note that the model-free property of our approach is particularly desirable since it is often unrealistic in practical settings that all the CTS subject to clustering are confined to one class of categorical models. In general, the distances are also robust against the choice of a nonoptimal set of lags. Based on previous considerations, it can be concluded that d_{CC} and d_B are two powerful and useful metrics to perform hard clustering of CTS.

Next section shows the behavior of the proposed dissimilarities in the fuzzy setting, which generally provides a greater deal of information than the crisp one.

4. Fuzzy clustering of categorical time series

So far, we have analysed the behavior of both d_{CC} and d_B in a crisp clustering context, where the collection of CTS is divided in C mutually exclusive groups. However, when dealing with time series, frequently a fuzzy clustering approach is more appealing and interpretable. It is not uncommon that some of the CTS in a particular set change their dynamics at a particular time, hence exhibiting patterns that share characteristics from several clusters. A meaningful clustering partition should disclose the vague nature of these elements, thus providing a better description of the temporal patterns of the series. In sum, the flexibility of the fuzzy logic in combination with the high ability of the proposed distances to discriminate between categorical processes motivates the use of both metrics in a soft clustering setting. In the present section, we first introduce the proposed fuzzy clustering algorithms and then carry out their assessment by means of several simulation experiments.

4.1. Fuzzy C-medoids clustering models based on the proposed dissimilarities

In this new framework, we attempt to perform clustering on the set of s categorical series $\mathscr{S} = \left\{X_t^{(1)}, \ldots, X_t^{(s)}\right\}$ by using the fuzzy C-medoids clustering model, which tries to find the subset of S of size C, $\widetilde{\mathscr{S}} = \left\{\widetilde{X}_t^{(1)}, \ldots, \widetilde{X}_t^{(C)}\right\}$, and the $s \times C$ matrix of fuzzy coefficients, $\boldsymbol{U} = (u_{ic}), i = 1, \ldots, s, c = 1, \ldots, C$, leading to the solution of the minimization problem

$$\min_{\widetilde{\mathscr{S}}, \boldsymbol{U}} \sum_{i=1}^{s} \sum_{c=1}^{C} u_{ic}^m d^*\left(X_t^{(i)}, \widetilde{X}_t^{(c)}\right) \text{ w.r.t. } \sum_{c=1}^{C} u_{ic} = 1, u_{ic} \geqslant 0, \tag{28}$$

where $u_{ic} \in [0, 1]$ represents the membership degree of the i-th CTS in the c-th cluster, d^* is a distance between CTS and $m > 1$ is a real number, usually referred to as fuzziness parameter, which regulates the fuzziness of the partition. For $m = 1$, the crisp version of the algorithm is obtained so that we have $u_{ic} = 1$ if the i-th series pertains to cluster c and $u_{ic} = 0$ otherwise. As the value of m increases, the boundaries between clusters get softer and the resulting partition is fuzzier.

To solve the minimization problem in (28), an iterative algorithm that alternately optimizes the membership degrees and the medoids is considered [38]. First, the membership degrees are optimized for a set of fixed medoids. The iterative solutions for the membership degrees are given by

$$u_{ic} = \left[\sum_{c\prime=1}^{C} \left(\frac{d^*\left(X_t^{(i)}, \widetilde{X}_t^{(c)}\right)}{d^*\left(X_t^{(i)}, \widetilde{X}_t^{(c\prime)}\right)}\right)^{\frac{1}{m-1}}\right]^{-1}, \tag{29}$$

for $i = 1, \ldots, s, c = 1, \ldots, C$.

In the second step, given the membership degrees computed according to (29), the C series minimizing (28) are selected as new medoids. This two-step mechanism is iterated until there is no change in the medoids or a maximum number of iterations is reached.

Two specific versions of the general problem in (28) are considered by taking into account $d^* = d_{CC}$ and $d^* = d_B$. An outline of the corresponding clustering algorithm is given in Algorithm 2.

Algorithm 2: The fuzzy C-medoids algorithm based on the proposed distances.

1: Fix C, m, *max.iter* and $d^* \in \{d_{CC}, d_B\}$
2: Set $iter = 0$
3: Pick the initial medoids $\widetilde{\mathscr{S}} = \left\{\widetilde{X}_t^{(1)}, \ldots, \widetilde{X}_t^{(C)}\right\}$
4: **repeat**
5: Set $\widetilde{\mathscr{S}}_{\text{OLD}} = \widetilde{\mathscr{S}}$ {Store the current medoids}
6: Compute $u_{ic}, i = 1, \ldots, s, c = 1, \ldots, C$, using (29)
7: For each $c \in \{1, \ldots, C\}$, determine the index $j_c \in \{1, \ldots, s\}$ satisfying:
$j_c = \text{argmin}_{1 \leqslant j \leqslant s} \sum_{i=1}^{s} u_{ic}^m d^*\left(X_t^{(i)}, X_t^{(j)}\right)$
8: **return** $\widetilde{X}_t^{(c)} = X_t^{(j_c)}$, for $c = 1, \ldots, C$ {Update the medoids}
9: $iter \leftarrow iter + 1$
10: **until** $\widetilde{\mathscr{S}}_{\text{OLD}} = \widetilde{\mathscr{S}}$ or $iter = max.iter$
11: **return** The resulting clustering partition.

4.2. Simulation study

A second simulation study was carried out to evaluate the performance of the fuzzy C-medoids clustering model based on d_{CC} and d_B. The new scenarios involve two well-separated clusters consisting of five time series each and a single isolated series arising from a different generating process. The specific scenarios and generating models are described below.

Scenario 4. Consider 3 three-state MC with corresponding transition matrices $\boldsymbol{P}_1^4, \boldsymbol{P}_2^4$ and $\boldsymbol{P}_3^4$ defined as

$$\begin{aligned} \boldsymbol{P}_1^4 &= Mat^3(0.1, 0.8, 0.1, 0.5, 0.4, 0.1, 0.6, 0.2, 0.2), \\ \boldsymbol{P}_2^4 &= Mat^3(0.2, 0.7, 0.1, 0.4, 0.3, 0.3, 0.2, 0.4, 0.4), \\ \boldsymbol{P}_3^4 &= \tfrac{1}{2}\left(\boldsymbol{P}_1^4 + \boldsymbol{P}_2^4\right). \end{aligned} \tag{30}$$

Scenario 5. Consider 3 three-state HMM with the same transition matrix, namely $\boldsymbol{P}^5 = \boldsymbol{P}_1^5 = \boldsymbol{P}_2^5 = \boldsymbol{P}_3^5$, but different emission matrices $\boldsymbol{E}_1^5, \boldsymbol{E}_2^5$ and $\boldsymbol{E}_3^5$ given by

$$\begin{aligned} \boldsymbol{P}^5 &= Mat^3(0.1, 0.8, 0.1, 0.5, 0.4, 0.1, 0.6, 0.2, 0.2), \\ \boldsymbol{E}_1^5 &= Mat^3(0.1, 0.8, 0.1, 0.5, 0.4, 0.1, 0.6, 0.2, 0.2), \\ \boldsymbol{E}_2^5 &= Mat^3(0.1, 0.4, 0.5, 0.25, 0.25, 0.5, 0.2, 0.5, 0.3), \\ \boldsymbol{E}_3^5 &= \tfrac{1}{2}\left(\boldsymbol{E}_1^5 + \boldsymbol{E}_2^5\right). \end{aligned} \tag{31}$$

Scenario 6. Consider 3 three-state NDARMA(2,0) models with marginal distribution $\boldsymbol{\pi}^6 = (1/3, 1/3, 1/3)$ and probabilities $(\phi_1, \phi_2, \varphi_0)_i^6, i = 1, 2, 3$, in the corresponding multinomial distribution given by

$$\begin{aligned} (\phi_1, \phi_2, \varphi_0)_1^6 &= (0.7, 0.15, 0.15) \\ (\phi_1, \phi_2, \varphi_0)_2^6 &= (0.1, 0.45, 0.45) \\ (\phi_1, \phi_2, \varphi_0)_3^6 &= \tfrac{(\phi_1, \phi_2, \varphi_0)_1^6 + (\phi_1, \phi_2, \varphi_0)_2^6}{2}. \end{aligned} \tag{32}$$

Note that Scenarios 4, 5 and 6 have been designed in a way that the isolated series is expected to lay "in the middle" of both clusters. In other words, a metric capable of discriminating between generating processes should be able to produce similar distance values from the isolated series to series from Cluster 1 and Cluster 2 indistinctly.

We considered $T \in \{200, 600\}$ and several values for the fuzziness parameter, namely $m = 1.5, 1.8, 2$ and 2.2. It is worth remarking that these values of m are coherent with the recommendations suggested by several authors [39–41]. The simulation mechanism was repeated 500 times. The number of clusters was set to $C = 2$. The computation of d_{CC} and d_B involved $\mathscr{L} = \{1\}$ in Scenarios 4 and 5, and $\mathscr{L} = \{1, 2\}$ in Scenario 6.

In this new setting, the clustering effectiveness was measured in terms of the number of times that the two resulting clusters were formed by 5 CTS coming from the same model and the isolated series presented relatively high membership degrees in both groups. To that end, we had to introduce a cutoff point to determine when a given realization is assigned to a specific cluster. According to the choice made in other works (e.g., [24,41,42]), the cutoff was set at 0.7, i.e., the i-th CTS was placed into the c-th cluster, $c = 1, 2$, if $u_{ic} > 0.7$. A discussion about the arguments supporting the choice of this threshold can be seen in [41]. In the same way, the isolated series was considered to concurrently pertain to both clusters if its membership degrees were both below 0.7. Note that, although d_{CZ} does not represent a fuzzy clustering approach, its evaluation can be done similarly by considering the final probabilities returned by the mixture model.

Note that the evaluation mechanism considered in Scenarios 4, 5 and 6 is particularly designed to handle fuzzy partitions, since it directly assesses the membership degrees of all the time series in the dataset. Particularly, a successful classification implies that the algorithm properly detects the vague nature of the isolated series. It is worth highlighting that several works have also considered scenarios with isolated series to evaluate fuzzy clustering algorithms (see, e.g., [24,25]).

The average classification rates attained by the analyzed dissimilarities in Scenarios 4, 5 and 6 are presented in Tables 11–13, respectively. In all cases, we decided to remove the results associated with distances d_{CZ}, d_{MV}, d_{DHM} and d_{OM} due to their poor performance. In fact, some of these metrics attained average rates of 0 in all the considered settings.

According to Table 11, the dissimilarity d_B was the most effective in Scenario 4 for every value of m and T, attaining significantly greater scores than its competitors. The metric d_{CC} achieved the worst rates of correct classification, but they were close to the ones associated with d_{MLE}, specially for $T = 600$, which suggests that no significant differences exist between both metrics in the considered setting.

A different situation occurs in Scenario 5. One can see from Table 12 that both proposed dissimilarities d_{CC} and d_B dramatically outperform the MLE-based metric for all combinations of m and T. The distance based on the binarized process seems to be slightly superior to its counterpart based on standard association measures when $T = 200$, but no significant differences are observed between them when $T = 600$.

Table 13 shows the results for Scenario 6. Here the metric d_B achieves the best classification rates for the shortest value of the series length. However, when increasing the value of T, the distance d_{CC} significantly improves its performance, reaching

Table 11
Average rates of correct classification for d_{CC}, d_B and d_{MLE} in Scenario 4. For each value of the fuzziness parameter and the series length, the best result is shown in bold.

	Distance	$m = 1.5$	$m = 1.8$	$m = 2.0$	$m = 2.2$
$T = 200$	d_{CC}	0.184	0.302	0.356	0.424
	d_B	**0.254**	**0.448**	**0.564**	**0.606**
	d_{MLE}	0.168	0.340	0.446	0.498
$T = 600$	d_{CC}	0.268	0.508	0.660	0.758
	d_B	**0.340**	**0.642**	**0.744**	**0.836**
	d_{MLE}	0.288	0.570	0.698	0.758

Table 12
Average rates of correct classification for d_{CC}, d_B and d_{MLE} in Scenario 5. For each value of the fuzziness parameter and the series length, the best result is shown in bold.

	Distance	$m = 1.5$	$m = 1.8$	$m = 2.0$	$m = 2.2$
$T = 200$	d_{CC}	0.316	0.528	0.620	0.676
	d_B	**0.380**	**0.562**	**0.666**	**0.708**
	d_{MLE}	0.070	0.072	0.046	0.032
$T = 600$	d_{CC}	0.449	0.675	**0.748**	**0.762**
	d_B	**0.494**	**0.688**	0.730	0.760
	d_{MLE}	0.120	0.104	0.076	0.052

Table 13
Average rates of correct classification for d_{CC}, d_B and d_{MLE} in Scenario 6. For each value of the fuzziness parameter and the series length, the best result is shown in bold.

	Distance	$m = 1.5$	$m = 1.8$	$m = 2.0$	$m = 2.2$
$T = 200$	d_{CC}	0.241	0.323	0.351	0.279
	d_B	**0.288**	**0.456**	**0.506**	0.508
	d_{MLE}	0.212	0.382	0.454	**0.542**
$T = 600$	d_{CC}	**0.442**	**0.748**	**0.835**	**0.875**
	d_B	0.366	0.634	0.770	0.872
	d_{MLE}	0.384	0.630	0.734	0.862

the highest scores for all values of m. The remaining dissimilarity d_{MLE} attains significantly worse results than the best-performing one for $m \in \{1.5, 1.8, 2\}$, and shows a similar behavior when $m = 2.2$.

In short, although the great effectiveness of d_{CC} and d_B in CTS clustering was already observed by carrying out hard cluster analysis, the results presented in this section also illustrate how the fuzzy nature of time series exhibiting intermediate properties between categorical models is properly detected by fuzzy algorithms based on both metrics.

5. Time consumption comparison

In order to assess the efficiency of the five dissimilarities analyzed throughout Sections 3 and 4, we recorded the runtime of the corresponding programs used for the experiments in Scenario 3. Specifically, given a metric and a value for T, we reported the CPU runtime spent in finishing the clustering task for the 500 simulation trials. The computer used to run the programs was a MacBook Pro with processor Quad-Core Intel Core i7, a speed of 2.9 GHz and a RAM memory of 16 GB. The programs were coded and executed in RStudio. The R version was 4.1.2.

The CPU runtime for the five dissimilarity measures is provided in Table 14. The more efficient distances were d_B and d_{CZ}, followed by d_{CC}. The metrics d_{MLE} and d_{MV} were substantially slow. For instance, both distances spent more than 10 and 20 h in finishing the clustering task for $T = 200$, respectively. Furthermore, all the dissimilarities run at most linearly with the series length except for d_{MV}, which exhibits a quadratic relationship. This makes d_{MV} the worst distance by far in terms of efficiency.

In summary, both metrics d_{CC} and d_B are efficient, but the latter significantly outperforms the former in terms of computational speed. In fact, d_B is the fastest among all the considered dissimilarities.

6. Applications. Clustering of biological sequences

This section is devoted to show two applications of the proposed clustering procedures. The first application considers DNA sequences from several viruses, while the second focuses on protein data from two different species. In both cases,

Table 14
The CPU runtime (minutes) for five metrics regarding the 500 simulation trials in Scenario 3.

Measure	$T = 200$	$T = 600$
d_{CC}	8.8538	22.7154
d_B	1.9062	2.6503
d_{MLE}	645.9365	1901.7660
d_{CZ}	2.6986	4.7045
d_{MV}	1275.6150	13134.4900

we first describe the corresponding database along with some exploratory analyses and, afterward, we show the results of applying the clustering algorithms.

6.1. Clustering of DNA sequences

This section shows the application of the proposed methods to a dataset of DNA sequences.

6.1.1. Dataset and exploratory analyses

In this case study, we consider the genome of 32 different viruses. The genome of an organism consists of sequences of the four DNA bases, the purines adenine ('a') and guanine ('g'), and the pyrimidines thymine ('t') and cytosine ('c'). Therefore, it can be seen as a CTS with range $\mathscr{V} = \{\mathtt{a}, \mathtt{g}, \mathtt{c}, \mathtt{t}\}$, containing all genetic information of the organism. For instance, the first symbols in one of the sequences of the considered database are given by the subsequence

atggcccaagcacaaattct...,

which corresponds to the virus *Rodent associated circovirus 1*. This type of CTS exhibit many signs of possible serial dependence. In fact, several authors have considered different types of categorical processes to model DNA sequences (see, e.g., [3,4,7]).

Each virus in the considered database pertain to one of four different families, so-called *Rodent associated circovirus* (RAC), *Circoviridae LDMD* (CLDMD), *Human cosavirus* (HC) and *Human parechovirus* (HP). In turn, the families RAC and CLDMD pertain to the subgroup of *Circoviridaes*, whereas the families HC and HP belong to the subgroup of *Picornaviridaes*. Table 15 summarizes information about the families, subgroups, and number of instances considered within each family. All data were sourced from the National Center for Biotechnology Information (NCBI) website[2]. The DNA sequences under consideration have different lengths, ranging from $T = 669$ (minimum length) to $T = 6950$ (maximum length), with a median value of $T = 988$. However, this is not an issue for the computation of d_{CC} and d_B, since these metrics do not require both time series to have the same length (see Remark 5).

According to Table 15, one could assume the existence of four different groups within the set of genetic sequences under consideration. Our objective is to determine if the proposed metrics d_{CC} and d_B are able to discover the existence of the underlying families, i.e., if the genetic families can be distinguished in terms of generating stochastic processes.

As a preliminary exploratory step, we performed a two-dimensional scaling (2DS) based on both pairwise dissimilarity matrices, those computed by using d_{CC} and d_B. That way, a projection of the genetic sequences on a two-dimensional plane preserving the original distances as well as possible is available. The set of lags $\mathscr{L} = \{1, 2, \ldots, 6\}$ was considered for the computation of both distances (see Section 6.1.2).

The location of the 32 sequences in the transformed space is displayed in Fig. 3. The left panel corresponds to the distance d_{CC}, whereas the right panel refers to the metric d_B. The R^2 values are 0.7677 and 0.8203, respectively. Thus, the scatter plots in Fig. 3 can be considered quite accurate representations of the underlying distance configurations [43]. The points in Fig. 3 have been colored according to the classes presented in Table 15 concerning the family of each virus.

It is clear from Fig. 3 that both distances are able to detect some structure in the dataset according to the underlying families of viruses. The left-hand panel shows a fuzzy configuration. Each class of viruses is located in a different region of the plane, but there are several points situated in the middle of the graph, which could pertain to more than one family. On the contrary, the right-hand panel displays a more compact conformation, with the families HC and HP clearly separated from each other and from families RAC and CLDMD. Although these latter classes show some degree of overlap, the corresponding groups are clearly distinguishable. Fig. 3 suggests that the distance d_B is more effective than d_{CC} if the goal is to obtain an accurate clustering partition according to the true families of viruses.

6.1.2. Application of clustering algorithms and results

The crisp and fuzzy clustering methods proposed in this work were applied to the dataset of genetic sequences. Concerning the hard methods, the number of clusters, C, and the set of lags, $\mathscr{L}$, must be determined in advance. The first hyperparameter was fixed to $C = 4$, since the considered viruses pertain to 4 different families. On the other hand, the heuristic

[2] https://www.ncbi.nlm.nih.gov/genome/browse#!/overview/.

Table 15
Summary of the 32 genetic sequences.

Family	Subgroup	No. instances
Rodent associated circovirus (RAC)	*Circoviridaes*	5
Circoviridae LDMD (CLDMD)	*Circoviridaes*	10
Human cosavirus (HC)	*Picornaviridaes*	10
Human parechovirus (HP)	*Picornaviridaes*	7

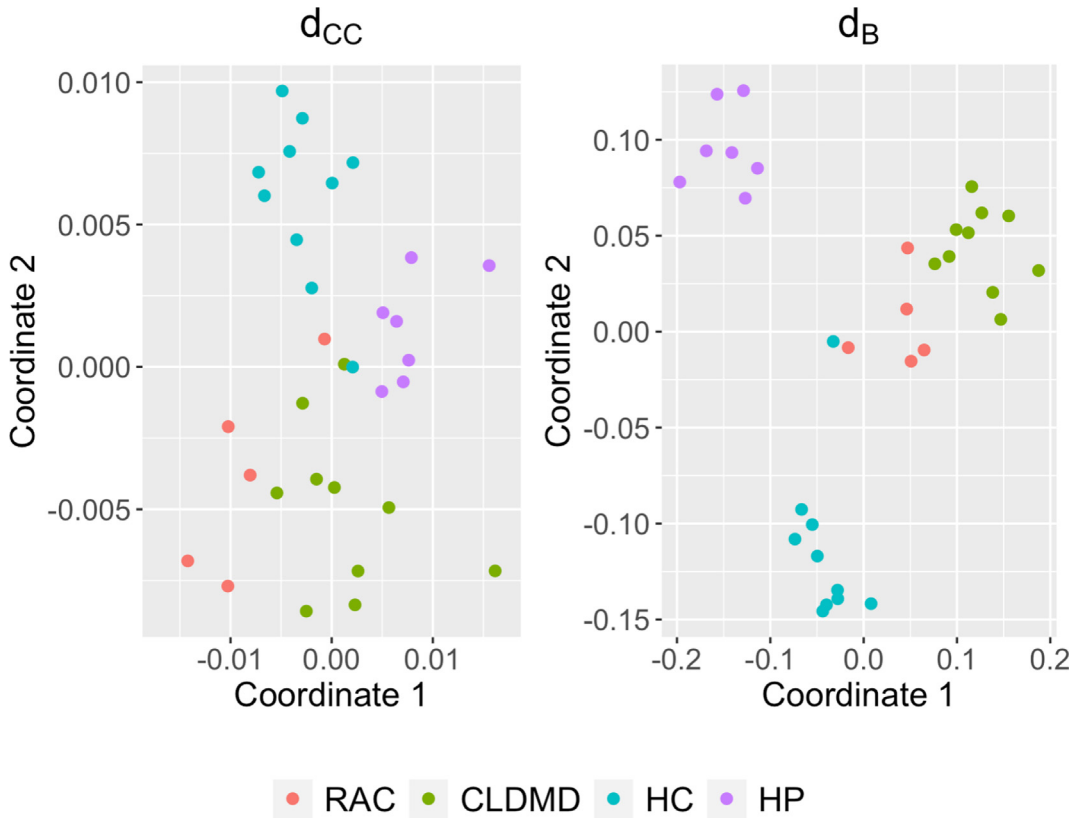

Fig. 3. Two-dimensional scaling plane based on distances d_{CC} (left panel) and d_B (right panel) for the 32 DNA sequences.

criterion presented in Section 3.4 (Steps 1–3) was employed to find out the optimal set of lags. Specifically, we fixed a global significance level $\alpha = 0.05$ and a maximum lag $L_{\text{Max}} = 10$, and recorded the most significant lag L_i for each genetic sequence. The collection $\mathscr{L} = \{1, 2, 3, 4, 5, 6\}$ was chosen as the optimal one. Note that the considered criterion can be used to select the set of lags in every application concerning the proposed methods.

As for the fuzzy clustering techniques, an additional hyperparameter, m, must be determined. The selection of this parameter was carried out by taking into account $C = 4$ and the optimal set of lags obtained with the crisp methods. In this regard, we run the fuzzy C-medoids algorithm based on both metrics for C and $\mathscr{L}$ fixed, and several values of $m, m \in \{1.1, 1.2, \ldots, 4\}$. The optimal value for m was the one associated with the lowest value of the Xie-Beni index [44]. It is worth remarking that several works have used criteria based on internal clustering quality indices to carry out the selection of the optimal value for m (see, e.g., [45]), which is the one producing the partition with the best trade-off between compactness and separation. This procedure resulted in $m = 1.4$ for d_{CC} and $m = 1.8$ for d_B.

Table 16 contains the values of the ARI, JI and FMI indices for the crisp clustering procedures based on the two metrics. In all cases, the ground truth was assumed to be given by the underlying families of viruses as indicated in Table 15. Both dissimilarities attained satisfactory values for the evaluation indices. However, as it was expected from Fig. 3, the distance d_B produced a more accurate partition than d_{CC}. In fact, the experimental solution associated with d_B is almost identical to the ground truth except for the fact that one virus pertaining to the family HC is placed within the cluster which contains HP viruses. The misclassified organism corresponds to the central blue point in the right panel of Fig. 3.

For comparison purposes, the results obtained by the distances d_{MLE} and d_{CZ} (the best-performing metrics among the alternative ones as indicated by the simulations carried out throughout the paper) are also shown in Table 16. The estimation

Table 16
Values of three clustering quality indices for several dissimilarities in the database of DNA sequences. For each index, the best result is shown in bold. The last column contains the average number of iterations.

Method	ARI	JI	FMI	Number of iterations
d_{CC}	0.665	0.597	0.748	2.616
d_B	**0.912**	**0.875**	**0.933**	2.092
$d_{MLE,MC}$	0.848	0.794	0.885	2.210
$d_{MLE,HMM}$	0.047	0.177	0.302	2.710
$d_{MLE,DAR}$	0.594	0.534	0.696	2.452
d_{CZ}	0	0.244	0.494	–

procedure in the former metric was carried out by considering independently a MC, a first-order HMM and a DAR(2) process. The notation $d_{MLE,MC}$, $d_{MLE,HMM}$ and $d_{MLE,DAR}$ was used in Table 16 to indicate the nature of the estimated coefficients. The best measure among the alternative distances was $d_{MLE,MC}$, which showed a worse but similar performance than d_B. The remaining metrics exhibited worse behavior. The problem with the distances based on estimated model coefficients is that, in practical applications like this, there is no way of knowing which model is the best for the CTS at hand, so it is impossible for the user to choose the optimal form of d_{MLE}.

The last column of Table 16 includes the average number of iterations until the PAM algorithm reached convergence. For each dissimilarity, 500 executions were performed and the same clustering partition was always obtained. The algorithm using distance d_B returned the clustering solution in approximately 2 iterations, whereas the number of iterations associated with the remaining approaches was slightly higher.

Table 17 contains the partition returned by the fuzzy C-medoids algorithm based on d_B. For each single virus, the entries in bold enhance the highest membership degrees, i.e, they provide the cluster assignment from a crisp perspective. The entries highlighted in italics in the first column correspond to the medoid viruses.

Table 17 provides a great deal of information about the corresponding partition, since nearly all the viruses in the collection display significant membership degrees in more than one group. The organisms in the family RAC exhibit the maximum membership degree in the same cluster, and the same happens for CLDMD and HP. On the other hand, the family HC is the most heterogeneous one. In fact, most of the viruses in this family show the maximum degree in the HP cluster. In fact, the

Table 17
Membership degrees for the 32 genetic sequences of viruses by considering the 4-cluster partition produced by the distance d_B.

Virus	C_1	C_2	C_3	C_4
RAC 1	0.188	0.108	**0.491**	0.213
RAC 2	0.209	0.126	**0.479**	0.186
RAC 3	0.000	0.000	**1.000**	0.000
RAC 4	0.212	0.136	**0.393**	0.259
RAC 5	0.259	0.208	**0.365**	0.168
CLDMD 1	0.000	0.000	0.000	**1.000**
CLDMD 2	0.215	0.124	0.223	**0.438**
CLDMD 3	0.237	0.144	0.236	**0.384**
CLDMD 4	0.256	0.150	0.216	**0.377**
CLDMD 5	0.191	0.113	0.214	**0.483**
CLDMD 6	0.177	0.100	0.191	**0.533**
CLDMD 7	0.202	0.105	0.178	**0.515**
CLDMD 8	0.266	0.157	0.181	**0.395**
CLDMD 9	0.259	0.168	0.260	**0.312**
CLDMD 10	0.223	0.137	0.258	**0.382**
HC 1	**0.339**	0.296	0.158	0.207
HC 2	**0.362**	0.345	0.151	0.143
HC 3	**0.356**	0.327	0.166	0.151
HC 4	0.276	**0.323**	0.214	0.187
HC 5	**0.346**	0.300	0.180	0.174
HC 6	0.308	**0.369**	0.171	0.152
HC 7	**0.324**	0.323	0.195	0.159
HC 8	**0.336**	0.332	0.172	0.159
HC 9	0.000	**1.000**	0.000	0.000
HC 10	**1.000**	0.000	0.000	0.000
HP 1	**0.346**	0.248	0.203	0.202
HP 2	**0.318**	0.208	0.198	0.276
HP 3	**0.368**	0.188	0.260	0.185
HP 4	**0.369**	0.229	0.220	0.182
HP 5	**0.324**	0.246	0.219	0.211
HP 6	**0.359**	0.241	0.201	0.199
HP 7	**0.396**	0.241	0.185	0.178

medoid of this cluster corresponds to a HC virus. These insights suggest that there is a high degree of overlap between HC and HP families.

Since the dissimilarities d_{CC} and d_B are only well-defined for stationary series, it is important to check the stationarity of the genetic sequences under consideration. To that aim, the so-called rate evolution graph [46] was constructed for each one of the medoid series associated with the d_B-based fuzzy procedure. This tool represents component-wise curves of the cumulated sums $\boldsymbol{C}_t = \sum_{s=1}^{t} \boldsymbol{Y}_t, t = 1, \ldots, T$, against time t. The slope of each graph gives an estimation for the corresponding marginal probability. If the underlying process is stationary, then each component should exhibit a linear behavior in t, whereas apparent violations of linearity reveal nonstationarity. Fig. 4 depicts the rate evolution graphs of the medoids. In all cases, no strong deviations from linearity are observed, which suggests a stationary behavior for the medoid series. The rate evolution graphs of the remaining sequences are similar to those in Fig. 4. In short, the stationarity of the series corroborates the suitability of both d_{CC} and d_B to perform clustering in the genetic database.

The conclusions reached from a fuzzy analysis like the previous one could be particularly helpful from a biological point of view. First, the medoid sequences could be used to describe the whole families. Note that the medoids represent the prototype features of the clusters, then summarizing the characteristics of the viruses within each group. For example, a researcher could employ the virus RAC 3 as a representative element of the RAC class, thus avoiding a joint analysis of all the organisms in the family. Second, the overlapping nature of some viruses may give some hints on their common evolutionary patterns. For example, it is clear that the families HC and HP are related, which is reasonable since viruses in both groups have human beings as natural hosts. One could even analyse the specific membership degrees (see Table 17) and determine that the virus HC 2 is the one displaying the highest degree of overlap between the HC and HP families. Note that, although the previous case study dealt with genomes of well-known viruses, the proposed methodology could be employed to cluster databases of recently discovered DNA sequences, which would shed light on the common features of the corresponding organisms. Third, the resulting clustering partition is useful in its own right, since it provides meaningful groups which may be helpful for exploratory purposes.

6.2. Clustering of protein sequences

This section shows the application of the proposed methods to a dataset of protein sequences.

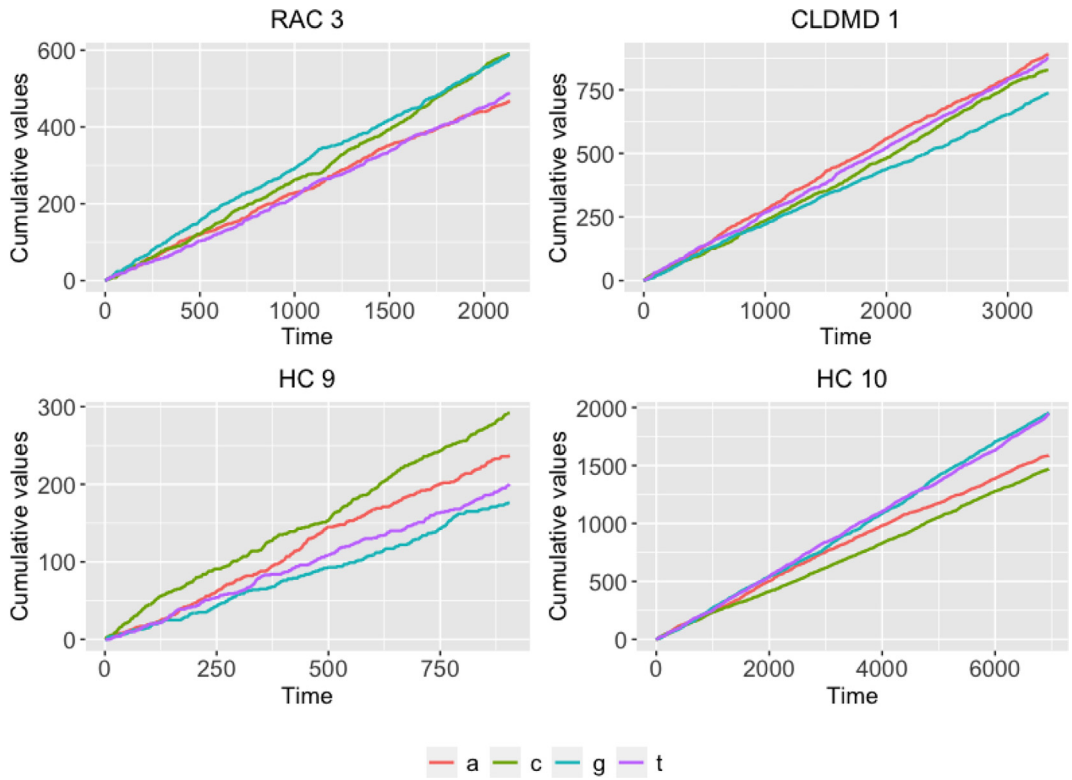

Fig. 4. Rate evolution graphs for the medoid genetic sequences.

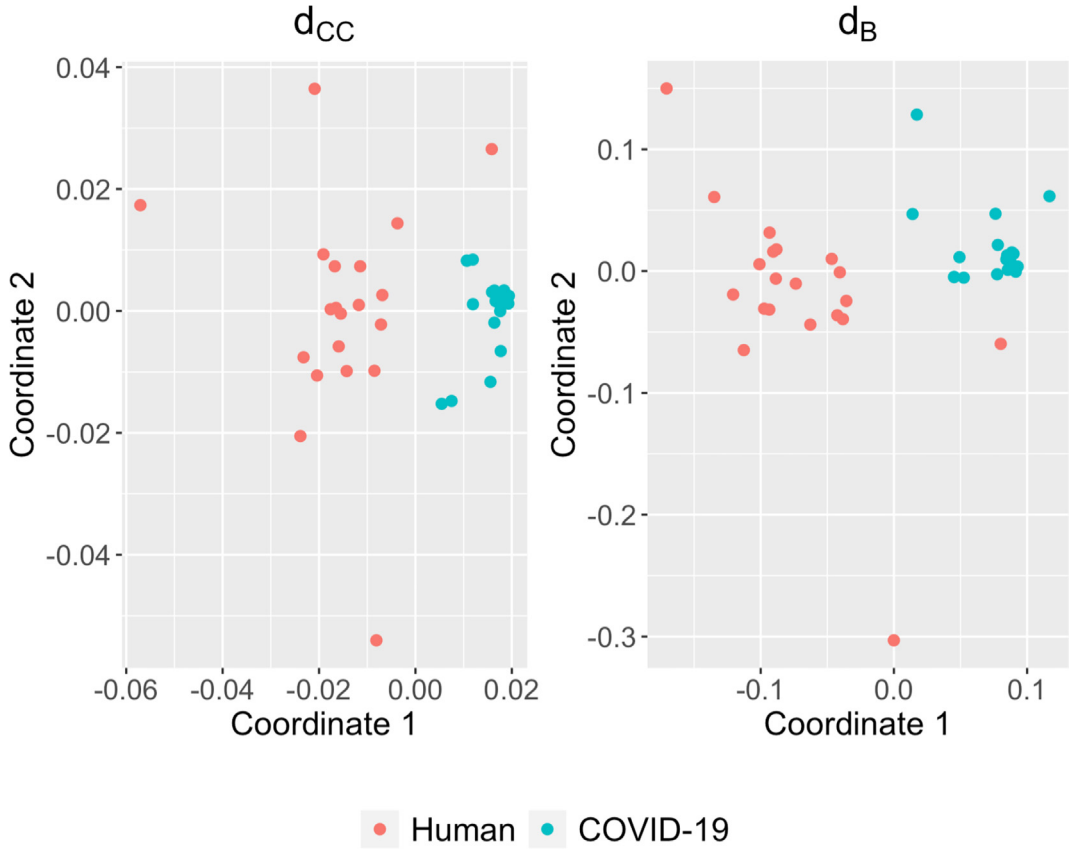

Fig. 5. Two-dimensional scaling plane based on distances d_{CC} (left panel) and d_B (right panel) for the 40 protein sequences.

6.2.1. Dataset and exploratory analyses

In this second application, we consider a dataset with 40 protein sequences. Proteins are large molecules constituted of one or more chains of simple organic compounds called amino acids. There are 20 different amino acids making up the proteins of any living organism. Therefore, each protein can be seen as a CTS with 20 categories. In fact, the application of categorical processes to protein data has been considered in several works [6,47].

Half of the proteins in the database are found in different parts of human beings, while the remaining half are present in several variants of COVID-19 virus. In this regard, our goal is to find out if the proposed clustering algorithms are able to group the sequences according to the underlying species. The maximum, minimum and median lengths for the CTS in the database are $T = 2511, T = 165$ and $T = 426$, respectively. The corresponding data were extracted from the UniProt Knowledgebase (UniProtKB) website.[3]

Note that, in this case study, a number of 400 features of the form $\widehat{p}_{ij}(l)$ or $\widehat{\phi}_{ij}(l)$ must be estimated from each CTS to compute d_{CC} and d_B, respectively. However, as we are dealing with realizations of moderate size, the corresponding quantities are expected to be poorly estimated. For this reason, we decided to reduce the number of categories in the original CTS by employing the so-called protein sequence encoding. Specifically, we chose to categorize the amino acids into three classes according to its hydrophobicity, which is a common transformation [48,49]. As result, the new dataset contains CTS with 3 categories, which allows to estimate features $\widehat{p}_{ij}(l)$ or $\widehat{\phi}_{ij}(l)$ with a higher degree of accuracy.

As in Section 6.1.1, a 2DS was carried out by considering each one of the proposed metrics. Computation of both dissimilarities was performed by using the set $\mathscr{L} = \{1, 2\}$, which was selected as the optimal one according to the method provided in Section 3.4 for $\alpha = 0.05$ and $L_{\text{Max}} = 10$. Fig. 5 contains the corresponding planes for d_{CC} and d_B, whose associated R^2 values are 0.9500 and 0.6517, respectively. Different colours were used to distinguish human proteins from COVID-19 proteins. Configurations in Fig. 5 indicate that both metrics are able to differentiate the underlying classes quite properly in this database.

[3] https://www.uniprot.org.

Table 18
Values of three clustering quality indices for several dissimilarities in the database of protein sequences. For each index, the best result is shown in bold. The last column contains the average number of iterations.

Method	ARI	JI	FMI	Number of iterations
d_{CC}	**0.900**	**0.903**	**0.949**	1
d_B	**0.900**	**0.903**	**0.949**	1
$d_{MLE,MC}$	0.715	0.745	0.854	1
$d_{MLE,HMM}$	0.344	0.504	0.670	1
$d_{MLE,DAR}$	0.805	0.818	0.900	1
d_{CZ}	0	0.487	0.698	–

6.2.2. Application of clustering algorithms and results

The proposed clustering algorithms were applied to the database of protein sequences. The number of clusters was set to $C = 2$, which is the number of underlying species in this new case study (human and COVID-19). Concerning the fuzzy techniques, the parameter m was selected by means of the procedure presented in Section 6.1.2, resulting $m = 2$ for d_{CC} and $m = 2.3$ for d_B.

Table 18 summarizes the performance of the hard clustering algorithms based on the proposed and the alternative metrics. Dissimilarities d_{CC} and d_B achieve the best results according to the three clustering quality indices. In fact, the clustering solution produced by both metrics is exactly the same. Specifically, one of the groups contains all the COVID-19 proteins along with one human protein, while the other cluster is formed by the remaining human proteins. The misclassified sequence corresponds to a protein identified as Q7Z434, which plays an important role in the human immune system. The MLE-based metric assuming NDARMA models also exhibits an excellent behavior in this application, since it returns a partition with only 2 misclassifications. The last column in Table 18 indicates that all methods converge in just one iteration.

Although the soft partitions produced by the fuzzy C-medoids algorithm based on d_{CC} and d_B are not shown for the sake of simplicity, they provide interesting insights. In both cases, COVID-19 proteins exhibit maximum membership degrees significantly higher than human proteins, which is coherent with the plots in Fig. 5, where the blue points constitute more compact clusters than the red ones. In addition, the partitions contain some sequences displaying a strongly fuzzy behavior. For instance, protein Q7Z434, which was incorrectly located in the COVID-19 group by the hard clustering procedures, has membership degrees of 0.37 and 0.63 in the human and COVID-19 clusters, respectively, according to dissimilarity d_B. This suggests that this protein shares a moderate degree of similarity with proteins in the former group, which is expected since Q7Z434 is actually present in human beings. Similar conclusions can be reached for several other sequences in the database.

7. Concluding remarks and future work

In this paper, we introduced two metrics to perform cluster analysis of categorical series. The goal of both distances is to discriminate between underlying categorical processes. The first dissimilarity (d_{CC}) is based on standard association measures, whereas the second (d_B) employs the so-called binarization of a categorical process, which describes the presence of each category by means of unit vectors. In both cases, the concepts of unsigned and signed dependence are properly assessed.

The proposed dissimilarities were used to define hard and soft clustering methods. To evaluate the performance of the constructed algorithms, a broad range of simulation scenarios covering a wide variety of categorical processes was taken into account. On the one hand, scenarios formed by CTS pertaining to well-defined clusters were considered to assess the hard clustering algorithms. On the other hand, scenarios involving series equidistant from two clusters were used to evaluate the fuzzy techniques. All numerical experiments showed the superiority of the proposed procedures in comparison with some alternative methods suggested in the literature. The computation times of the different techniques were also studied. The algorithms based on d_B achieved the best overall results in terms of both clustering effectiveness and computational efficiency.

To illustrate the usefulness of the novel categorical distances, we applied the corresponding clustering algorithms to two collections of biological sequences from different species. In both cases, the hard clustering methods based on both d_{CC} and d_B were able to detect the underlying structures, the latter attaining almost the highest possible accuracy. In addition, the fuzzy clustering methods provided a great deal of information, showing that some organisms share a certain degree of overlap. The results suggest that applied researchers in fields related to Biology and Genetics could substantially benefit from the proposed methodology when dealing with databases arising in those fields.

There are several ways through which this work can be expanded. First, the proposed dissimilarities could be extended so that they are able to discriminate also between different geometric profiles. In this way, metrics simultaneously taking into account serial dependence and proximity between raw categorical values could be introduced. Second, although we have presented here a simple but accurate technique for choosing the set of lags in the computation of both d_{CC} and d_B (see Section 3.4), the selection of the optimal set $\mathscr{L}$ could be a topic for further research. Specifically, this problem could be addressed through two different approaches: (i) by considering a feature selection problem in an unsupervised setting but assuming

that the search is limited to specific groups of features (note that each lag $l \in \mathbb{Z}$ produces a certain set of features for both d_{CC} and d_B) and (ii) by incorporating in the objective function of the clustering algorithms a set of weights giving different importance to each lag $l \in \mathbb{Z}$, which allows the ability of each lag to discriminate between groups to be automatically determined during the computation of the clustering partition. Third, robust clustering methods based on the proposed dissimilarities could be constructed by considering the so-called metric, noise and trimmed approaches [50]. Fourth, the statistical properties of d_{CC} and d_B could be investigated in order to define powerful hypothesis tests. In this way, clustering methods based on the p-value of the corresponding tests could be designed. Finally, given the great results obtained by the clustering algorithms in the biological datasets, they could be applied to complete biological databases in order to detect complex relationships between a great number of species. This could provide meaningful insights from an evolutionary perspective. All paths will be properly addressed in further work.

CRediT authorship contribution statement

Á**ngel Ló**p**ez-Oriona:** Conceptualization, Writing - review & editing, Methodology, Software, Visualization. **José A. Vilar:** Conceptualization, Supervision, Writing - review & editing, Project administration. **Pierpaolo D'Urso:** Conceptualization, Supervision, Writing - review & editing, Project administration.

Data availability

No data was used for the research described in the article.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors are grateful to the anonymous referees for their comments and suggestions. The research of Ángel López-Oriona and José A. Vilar has been supported by the Ministerio de Economía y Competitividad (MINECO) grants MTM2017-82724-R and PID2020-113578RB-100, the Xunta de Galicia (Grupos de Referencia Competitiva ED431C-2020-14), and the Centro de Investigación del Sistema Universitario de Galicia "CITIC" grant ED431G 2019/01; all of them through the European Regional Development Fund (ERDF). This work has received funding for open access charge by Universidade da Coruña/CISUG. The author Ángel López-Oriona is very grateful to researcher Maite Freire for her lessons about DNA theory.

References

[1] S. Rani, G. Sikka, Recent techniques of clustering of time series data: Asurvey, Int. J. Comput. Appl. 52 (15) (2012) 1–9.
[2] E. Maharaj, P. D'Urso, J. Caiado, Time Series Clustering and Classification, CRC Press, Chapman & Hall/CRC Computer Science and Data Analysis Series, 2019.
[3] K. Fokianos, B. Kedem, Regression theory for categorical time series, Stat. Sci. 18 (3) (2003) 357–376.
[4] C.H. Weiss, R. Gob, Measuring serial dependence in categorical time series, AStA-Adv. Stat. Anal. 92 (1) (2008) 71–89.
[5] D.S. Stoffer, D.E. Tyler, D.A. Wendt, The spectral envelope and its applications, Stat. Sci. 224–253 (2000).
[6] A. Krogh, M. Brown, I.S. Mian, K. Sjölander, D. Haussler, Hidden markov models in computational biology: Applications to protein modeling, J. Mol. Biol. 235 (5) (1994) 1501–1531.
[7] C.H. Weiß, An introduction to discrete-valued time series, John Wiley & Sons, 2018.
[8] I. Cadez, D. Heckerman, C. Meek, P. Smyth, S. White, Model-based clustering and visualization of navigation patterns on a web site, Data Min. Knowl. Discov. 7 (4) (2003) 399–424.
[9] C. Pamminger, S. Frühwirth-Schnatter, Model-based clustering of categorical time series, Bayesian Anal. 5 (2) (2010) 345–368.
[10] S. Frühwirth-Schnatter, C. Pamminger, R. Winter-Ebmer, A. Weber, Model-based clustering of categorical time series with multinomial logit classification, AIP Conf. Proc. 1281 (1) (2011) 1897–1900.
[11] J.G. Dias, Model selection criteria for model-based clustering of categorical time series data: A monte carlo study, in: Advances in data analysis, Springer, 2007, pp. 23–30.
[12] G.S., G.F., M.A., Clustering multivariate time series using hidden markov models, Int. J. Environ. Res. Public Health 11(3) (2014) 2741–2763. doi: 10.3390/ijerph110302741.
[13] T.F. Liao, D. Bolano, C. Brzinsky-Fay, B. Cornwell, A.E. Fasang, S. Helske, R. Piccarreta, M. Raab, G. Ritschard, E. Struffolino, M. Studer, Sequence analysis: Its past, present, and future, Soc. Sci. Res. 107 (2022), https://doi.org/10.1016/j.ssresearch.2022.102772 102772.
[14] C.H. Elzinga, Sequence analysis: Metric representations of categorical time series, Sociological methods and research.
[15] L. Lesnard, Setting cost in optimal matching to uncover contemporaneous socio-temporal patterns, Sociol. Methods. Res. 38 (3) (2010) 389–419.
[16] B. Halpin, Optimal matching analysis and life-course data: The importance of duration, Sociol. Methods. Res. 38 (3) (2010) 365–388.
[17] M. Studer, G. Ritschard, What matters in differences between life trajectories: a comparative review of sequence dissimilarity measures, J.R. Stat. Soc. Ser. A-Stat. Soc. 179 (2) (2016) 481–511.
[18] B. Halpin, Sadi: Sequence analysis tools for stata, Stata J. 17 (3) (2017) 546–572.
[19] M. García-Magariños, J.A. Vilar, A framework for dissimilarity-based partitioning clustering of categorical time series, Data Min. Knowl. Discov. 29 (2) (2015) 466–502.
[20] V. Melnykov, Clickclust: An r package for model-based clustering of categorical sequences, J. Stat. Softw. 74 (9) (2016) 1–34.

[21] A. Gabadinho, G. Ritschard, N.S. Müller, M. Studer, Analyzing and visualizing state sequences in R with TraMineR, J. Stat. Softw. 40 (4) (2011) 1–37, https://doi.org/10.18637/jss.v040.i04.
[22] Z. Huang, M. Ng, A fuzzy k-modes algorithm for clustering categorical data, IEEE Trans. Fuzzy Syst. 7 (4) (1999) 446–452.
[23] R. Krishnapuram, A. Joshi, O. Nasraoui, L. Yi, Low-complexity fuzzy relational clustering algorithms for web mining, IEEE Trans. Fuzzy Syst. 9 (2001) 595–607.
[24] P. D'Urso, E.A. Maharaj, Autocorrelation-based fuzzy clustering of time series, Fuzzy Sets Syst. 160 (24) (2009) 3565–3589.
[25] J.A. Vilar, B. Lafuente-Rego, P. D'Urso, Quantile autocovariances: a powerful tool for hard and soft partitional clustering of time series, Fuzzy Sets Syst. 340 (2018) 38–72.
[26] B. Lafuente-Rego, J.A. Vilar, Clustering of time series using quantile autocovariances, Adv. Data Anal. Classif. 10 (3) (2016) 391–415.
[27] Á. López-Oriona, J.A. Vilar, Quantile cross-spectral density: A novel and effective tool for clustering multivariate time series, Expert Syst. Appl. 185 (2021) 115677.
[28] J. Caiado, N. Crato, D. Peña, A periodogram-based metric for time series classification, Comput. Stat. Data Anal. 50 (10) (2006) 2668–2684.
[29] P. D'Urso, L. De Giovanni, R. Massari, R.L. D'Ecclesia, E.A. Maharaj, Cepstral-based clustering of financial time series, Expert Syst. Appl. 161 (2020) 113705.
[30] X. Wang, K. Smith, R. Hyndman, Characteristic-based clustering for time series data, Data Min. Knowl. Discov. 13 (3) (2006) 335–364.
[31] C.H. Weiss, Serial dependence of ndarma processes, Comput. Stat. Data Anal. 68 (2013) 213–238.
[32] C.H. Weiss, Empirical measures of signed serial dependence in categorical time series, J. Stat. Comput. Simul. 81 (4) (2011) 411–429.
[33] L. Kaufman, P.J. Rousseeuw, Finding groups in data: an introduction to cluster analysis, Vol. 344, John Wiley & Sons, 2009.
[34] A. Ahmad, L. Dey, A k-mean clustering algorithm for mixed numeric and categorical data, Data Knowl. Eng. 63 (2) (2007) 503–527.
[35] L. Hubert, P. Arabie, Comparing partitions, J. Classif. 2 (1) (1985) 193–218.
[36] E.B. Fowlkes, C.L. Mallows, A method for comparing two hierarchical clusterings, J. Am. Stat. Assoc. 78 (383) (1983) 553–569.
[37] A. Emdadi, F.A. Moughari, F.Y. Meybodi, C. Eslahchi, A novel algorithm for parameter estimation of hidden markov model inspired by ant colony optimization, Heliyon 5 (3) (2019) e01299.
[38] C. Döring, M.-J. Lesot, R. Kruse, Data analysis with fuzzy clustering methods, Comput. Stat. Data Anal. 51 (1) (2006) 192–214.
[39] R.L. Cannon, J.V. Dave, J.C. Bezdek, Efficient implementation of the fuzzy c-means clustering algorithms, IEEE Trans. Pattern Anal. Mach. Intell. 2 (1986) 248–255.
[40] J.C. Bezdek, Pattern recognition with fuzzy objective function algorithms, Springer Science & Business Media, 2013.
[41] E.A. Maharaj, P. D'Urso, Fuzzy clustering of time series in the frequency domain, Inf. Sci. 181 (7) (2011) 1187–1211.
[42] P. D'Urso, E.A. Maharaj, Wavelets-based clustering of multivariate time series, Fuzzy Sets Syst. 193 (2012) 33–61.
[43] J. Hair, R. Anderson, B. Babin, Multivariate Data Analysis, 7th Edition., Prentice Hall, 2009.
[44] X.L. Xie, G. Beni, A validity measure for fuzzy clustering, IEEE Trans. Pattern Anal. Mach. Intell. 13 (8) (1991) 841–847.
[45] Á. López-Oriona, J.A. Vilar, P. D'Urso, Quantile-based fuzzy clustering of multivariate time series in the frequency domain, Fuzzy Sets Syst. 443 (2022) 115–154, from Learning to Modeling and Control.
[46] R.L. Ribler, Visualizing categorical time series data with applications to computer and communications network traces Ph.D. thesis, Virginia Polytechnic Institute and State University, 1997.
[47] G. Wu, Frequency and markov chain analysis of amino acid sequences of mouse p53, Hum. Exp. Toxicol. 19 (9) (2000) 535–539.
[48] I. Dubchak, I. Muchnik, S.R. Holbrook, S.-H. Kim, Prediction of protein folding class using global description of amino acid sequence, Proc. Nat. Acad. Sci. 92 (19) (1995) 8700–8704.
[49] I. Dubchak, I. Muchnik, C. Mayor, I. Dralyuk, S.-H. Kim, Recognition of a protein fold in the context of the scop classification, Proteins: Structure, Function, and Bioinformatics 35 (4) (1999) 401–407.
[50] Ángel López-Oriona et al, Quantile-based fuzzy C-means clustering of multivariate time series: Robust techniques, International Journal of Approximate Reasoning 150 (2022) 55–82.