11-1-2022

Clustering mixed-type data using a probabilistic distance algorithm[Formula presented]

Cristina Tortora
San Jose State University, cristina.tortora@sjsu.edu

Francesco Palumbo
Università degli Studi di Napoli Federico II

Recommended Citation

Cristina Tortora and Francesco Palumbo. "Clustering mixed-type data using a probabilistic distance algorithm[Formula presented]" *Applied Soft Computing* (2022). https://doi.org/10.1016/j.asoc.2022.109704

Applied Soft Computing 130 (2022) 109704

Contents lists available at ScienceDirect

Applied Soft Computing

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

Clustering mixed-type data using a probabilistic distance algorithm

Cristina Tortora [a,*], Francesco Palumbo [b]

[a] San José Sate University, One Washington Square, San José, 95192, CA, USA
[b] University of Naples Federico II, Corso Umberto I, Napoli, 80138, Italy

ARTICLE INFO

Article history:
Received 7 March 2022
Received in revised form 30 September 2022
Accepted 4 October 2022
Available online 14 October 2022

MSC:
62
07

Keywords:
Probabilistic distance clustering
Mixed-type data
Fuzzy clustering

ABSTRACT

Cluster analysis is a broadly used unsupervised data analysis technique for finding groups of homogeneous units in a data set. Probabilistic distance clustering adjusted for cluster size (PDQ), discussed in this contribution, falls within the broad category of clustering methods initially developed to deal with continuous data; it has the advantage of fuzzy membership and robustness. However, a common issue in clustering deals with treating mixed-type data: continuous and categorical, which are among the most common types of data. This paper extends PDQ for mixed-type data using different dissimilarities for different kinds of variables. At first, the PDQ for mixed-type data is defined, then a simulation design shows its advantages compared to some state of the art techniques, and ultimately, it is used on a real data set. The conclusion includes some future developments.

Code metadata

Permanent link to reproducible Capsule: https://doi.org/10.24433/CO.5531267.v1

1. Introduction

Data clustering refers to a vast range of numerical algorithms designed to find groups of homogeneous data using systematic numerical methods. If the number of scientific papers devoted to reviewing clustering algorithms in specific and general domains is assumed as a proxy variable measuring the interest towards clustering, then launching the Google Scholar query {`review clustering`}, the query would return more than 350 entries, limiting the search to only the articles' titles. However, despite all, a unique definition for data clustering would not be found because diverse definitions and formalization exist in each research domain intended to be maximally adherent to what data clustering in that specific domain represents. In the last few decades, the numerical and computational perspective has become even more crucial because of the ever more considerable amount of available data.

Two approaches for data clustering exist: hierarchical and non-hierarchical (e.g., see [1]). The former leads to an indexed hierarchy through an agglomerating or divisive process. In contrast, the latter exploits iterative algorithms pursuing a statistical or geometrical homogeneity criterion to partition the data into a given number of clusters. In general, non-hierarchical algorithms' complexity is linearly dependent on the total number of statistical units, and they are easily parallelizable; therefore, they can be considered a practical and helpful approach to clustering large and even huge data sets [2]. Moreover, under the hypothesis that the data are generated from a mixture of known multivariate distributions with unknown vectors of parameters, the clustering problem can be afforded by estimating the mixture's parameters [3,4]; these algorithms are referred to as model-based clustering. Geometrical algorithms, instead, optimize a homogeneity criterion, where the homogeneity is measured in terms of distance among statistical units. Model-based clustering provides estimations with desirable statistical properties from an inferential point of view; however, it requires that the distributional hypotheses are satisfied. Moreover, model-based clustering might have more convergence issues than geometrical approaches [5, sect. 3.6]. In both cases, most algorithms determine two quantities: cluster memberships and cluster parameters. The computation of parameters and memberships depend on one another; therefore, the problem cannot be solved with direct optimization [6, Chap. 9]. The solution is determined through an iterative algorithm that alternatively computes the two quantities and stops when the optimized criterion reaches a local or a global minimum (maximum). The membership can be *crisp*, where a point belongs or does not belong to a given cluster, or *probabilistic* where a point can be assigned to one or more clusters with degrees of probability.

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail address: cristina.tortora@sjsu.edu (C. Tortora).

https://doi.org/10.1016/j.asoc.2022.109704

This article focuses on the Probability Distance Clustering (PDC) [7] and its extension to the mixed-data type, which falls in the geometrical data clustering approaches with probabilistic cluster membership.

As first introduced by Iyigun [8], and Ben-Israel and Iyigun [7], PDC is an iterative two-step clustering algorithm that takes into input the number of groups and alternately updates the probabilistic memberships and the centers. It optimizes a *classifiability* criterion called joint distance function (JDF) that depends on the distance between every point from all centers. More recently, Tortora [9] and Tortora et al. [10] proposed factor PDC (FPDC), a generalization of PDC in a reduced space. Based on Tucker3 [11, 12], FPDC exploits Tucker3 decomposition to obtain a subspace consistent with the PDC criterion; this extension is handy for large data sets [13]. Another proposed generalization introduces a new notion of dissimilarity that is grounded on a generic multivariate density function and increases the flexibility in cluster shapes [14]. PDC ensures a convex target function, and the overall JDF criterion decreases or at least does not increase at each step. The algorithm stops when JDF does not vary from one step to the next.

Most clustering approaches, including PDC in Ben-Israel and Iyigun's original proposal, aim to optimize a criterion that involves just one type of variable. However, in our multi-facet world, data stored in the databases of companies and institutions consist of more than one type of variable, e.g., data that refer to customers or goods or patients consist of categorical and continuous variables. One possible solution is to re-code mixed data into a single data type by, for example, transforming the continuous variables into categorical ones [15]. Although widely appreciated, this approach requires data pre-processing, such that the original association structure can result in significantly weakened. Several clustering methods specific to mixed data exist; for example, a good model-based method was proposed by McParland and Gormley [16]; Mbuga and Tortora [17] recently proposed a method based on a graph-based clustering technique. Although both techniques give excellent clustering results, they tend to be slow. The primary issue in clustering mixed data is the substantial gap between the similarity metrics in numerical and categorical data needing to identify a unified similarity metric [18]. Some approaches have been created based on this idea; among those techniques, the most common are *k*-prototypes [19] and KAy-means for MIxed LArge data (Kamila) [20]. For more detailed reviews on mixed-type data clustering, see [21–23].

This paper proposes an extension of PDC for mixed-type data within this framework. Since PDC is based on a dissimilarity matrix, a suitable newly defined dissimilarity measure, integrating different variables, is proposed. The cluster parameters that optimize the criterion based on the updated dissimilarity are then found and integrated into the algorithm.

2. PD-clustering algorithm

Let $\mathbf{X}$ be a data matrix with n units and J variables, and consider K (non-empty) clusters, with K assumed to be *a priori* known, probabilistic distance (PD) clustering [7] aims to find homogeneous clusters in the data according to two quantities: the distance of each data point $\boldsymbol{x}_i$ from each cluster center $\mathbf{c}_k$, denoted as d_{ik}, and the probability of each point belonging to a cluster, i.e., p_{ik}, for $k = 1, \ldots, K$ and $i = 1, \ldots, n$. PD-clustering relies on the following expression

$$p_{ik}d_{ik} = F(\boldsymbol{x}_i),$$

stating that for any $\boldsymbol{x}_i$ the product of the distance d_{ik} and the probability p_{ik} is a constant denoted by $F(\boldsymbol{x}_i)$, for $k = 1, \ldots, K$ [7]. $F(\boldsymbol{x}_i)$ is a constant for each observation and, therefore, does not depend on k. As the distance from the cluster center decreases, the probability of the point belonging to the cluster increases. The quantity $F(\boldsymbol{x}_i)$ is impacted by the closeness of $\boldsymbol{x}_i$ to the cluster centers, and it measures the classifiability of the point $\boldsymbol{x}_i$ to the K centers $\mathbf{c}_k$, for $k = 1, \ldots, K$. The smaller the $F(\boldsymbol{x}_i)$, the higher the probability of the point belonging to one cluster. If all of the distances between the point $\boldsymbol{x}_i$ and the centers of the clusters are equal to d_i, then $F(\boldsymbol{x}_i) = d_i/K$ and all of the probabilities of belonging to each cluster are equal, i.e., $p_{ik} = 1/K$. The sum of $F(\boldsymbol{x}_i)$ over i is called joint distance function (JDF)

$$JDF = \sum_{i=1}^{n}\sum_{k=1}^{K} p_{ik}d_{ik}.$$

The K centers that minimize the JDF maximize the overall classifiability. To account for clusters of different size, Iyigun and Ben Israel [24] proposed an extension adjusted for cluster size, namely PDQ clustering. The JDF is weighted as follow

$$JDF = \sum_{i=1}^{n}\sum_{k=1}^{K} \frac{p_{ik}d_{ik}}{q_k},$$

where q_k is the cluster size, under the constraint that $\sum_{k=1}^{K} q_k = n$. The p_{ik} can then be computed via

$$p_{ik} = \frac{\prod_{m\neq k} d_{im}/q_m}{\sum_{r=1}^{K}\prod_{m\neq r} d_{im}/q_m}. \tag{1}$$

Since $0 \leq p_{ik} \leq 1$ by definition then p_{ik}^2 is a monotonic decreasing transformation that preserves the optimal minimum. To allow for optimization through a quadratic form, Iyigun and Ben-Israel [24] proposed to find the cluster size q_k and the centers that maximize the classifiability minimizing the following adjusted *JDF* function

$$JDF = \sum_{i=1}^{n}\sum_{k=1}^{K} \frac{p_{ik}^2 d_{ik}}{q_k}.$$

The cluster size becomes

$$q_k = n\frac{\left(\sum_{i=1}^{N} d_{ik}p_{ik}^2\right)^{1/2}}{\sum_{k=1}^{K}\left(\sum_{i=1}^{N} d_{ik}p_{ik}^2\right)^{1/2}}, \tag{2}$$

for $k = 1, \ldots, K-1$, and

$$q_K = n - \sum_{k=1}^{K-1} q_k.$$

The optimal centers for continuous data using the Euclidean distance are

$$\mathbf{c}_k = \frac{\sum_{i=1}^{n}\frac{p_{ik}^2}{d_{ik}}\boldsymbol{x}_i}{\sum_{i=1}^{n}\frac{p_{ik}^2}{d_{ik}}}. \tag{3}$$

Further extensions of PDC exist; Tortora et al. [13] proposed a factor version of the method to deal with high-dimensional data. Recently, [14,25] further extended the method to include greater flexibility.

3. PD-clustering for mixed-type data

The present article proposes using Gower's dissimilarity [26] to deal with mixed-type variables in the PDQ algorithm. Let us define with $\mathbf{X}$ the $n \times J$ data matrix of a general element x_{ij} and with $\mathbf{C}$ the $K \times J$ matrix of centers; we assume that both $\mathbf{X}$ and $\mathbf{C}$ have L continuous variables, O ordinal variables, and M categorical variables. Binary variables are treated as categorical

ones. The Gower's dissimilarity between a generic data point $\mathbf{x}_i$ and a center $\mathbf{c}_k$ is:

$$D_{ik} = \frac{\sum_{j=1}^{J} w_{ikj} d_{ikj}}{\sum_{j=1}^{J} w_{ikj}},$$

where d_{ikj} is the dissimilarity between the observation data point $\mathbf{x}_i$ and the center $\mathbf{c}_k$ for the jth variable. It is worth noting that the quantity d_{ikj} is determined by a different expression, as described in the following, according to the type of variable. Let us define with $l = 1, \ldots, L$ the continuous variables, $o = 1, \ldots, O$ the ordinal variables, and $m = 1, \ldots, M$ the categorical variables, with $L + O + M = J$. For ordinal data, the dissimilarity measurement of choice is:

$$d_{iko} = \frac{|x_{io} - c_{ko}|}{R_o},$$

where d_{iko} is scaled between $0 \leq d_{iko} \leq 1$ by the division of R_o, the range of variable o. To be consistent with PDQ, the Euclidean distance is used for the L continuous variables, but it needs to be scaled. Scaling a distance is challenging, and different approaches can be used; Milligan and Cooper [27] suggest several approaches, an adaptation of one of the approaches works for this problem, and the distance is therefore defined as

$$d_{ik} = \sqrt{\sum_{l=1}^{L} \left(\frac{x_{il} - c_{kl}}{x_l^*} \right)^2},$$

where $x_l^* = 1$ if $-0.1 < \bar{x}_l < 0.1$, $x_l^* = \bar{x}_l$ otherwise. For binary data and categorical data, the exact match indicator function is usually used:

$$d_{ikm} = \begin{cases} 0 & x_{im} = c_{km} \\ 1 & x_{im} \neq c_{km}. \end{cases}$$

The weights traditionally used for Gower's dissimilarity are 1 for variables that are comparable and 0 for non-comparable variables or missing data. Weights proportional to the number of continuous, ordinal, and categorical variables were the most natural choice and appeared to cluster the best. The Gower's dissimilarity used is as follows:

$$D_{ik} = \frac{1}{J} \left(L \sqrt{\sum_{l=1}^{L} \left(\frac{x_{il} - c_{kl}}{\bar{x}_l} \right)^2} + O \frac{\sum_{o=1}^{O} |x_{io} - c_{ko}|}{R_o} + M \sum_{m=1}^{M} \mathbb{1}_{(x_{im} \neq c_{km})} \right), \tag{4}$$

where $L + O + M = J$ and the center of cluster k has been partitioned into $\mathbf{c}_k = [\mathbf{c}_{kL}, \mathbf{c}_{kO}, \mathbf{c}_{kM}]$, corresponding to the continuous, ordinal and categorical variables respectively.

3.1. Center updates

Using Gower's dissimilarity, the PDQ objective becomes to minimize

$$JDF = \frac{1}{J} \sum_{i=1}^{n} \sum_{k=1}^{K} \frac{p_{ik}^2}{q_k} \left(L \sqrt{\sum_{l=1}^{L} \left(\frac{x_{il} - c_{kl}}{\bar{x}_l} \right)^2} + O \frac{\sum_{o=1}^{O} |x_{io} - c_{ko}|}{R_o} + M \sum_{m=1}^{M} \mathbb{1}_{(x_{im} \neq c_{km})} \right).$$

To obtain the centers that minimize the JDF each type of variable can be considered separately. Let us start considering the continuous variables,

$$\frac{\partial JDF}{\partial \mathbf{c}_{kL}} = \frac{\partial}{\partial \mathbf{c}_{kL}} \frac{L}{J} \sum_{i=1}^{n} \sum_{k=1}^{K} \frac{p_{ik}^2}{q_k} \sqrt{\sum_{l=1}^{L} \left(\frac{x_{il} - c_{kl}}{\bar{x}_l} \right)^2}.$$

Setting the derivative equal to zero,

$$\frac{\partial}{\partial \mathbf{c}_{kL}} \sum_{i=1}^{n} \sum_{k=1}^{K} \frac{p_{ik}^2}{q_k} \sqrt{\sum_{l=1}^{L} \left(\frac{x_{il} - c_{kl}}{\bar{x}_l} \right)^2} = 0,$$

the optimization problem reduces to the same optimization problem as PDQ for continuous data using Euclidean distance; thus, the centers for continuous variables correspond to (3).

For the ordinal data, the derivatives reduce to be coordinate by coordinate

$$\frac{\partial JDF}{\partial c_{ko}} = \frac{\partial}{\partial c_{ko}} \frac{O}{J} \sum_{i=1}^{n} \sum_{k=1}^{K} \frac{p_{ik}^2}{q_k} \sum_{o=1}^{O} \frac{|x_{io} - c_{ko}|}{R_o},$$

and setting the derivatives equal to zero

$$\sum_{i=1}^{n} \frac{p_{ik}^2}{q_k} \frac{|x_{io} - c_{ko}|}{R_o} = 0,$$

we obtain

$$c_{ko} = \sum_{i=1}^{n} \frac{p_{ik}^2 x_{io}}{\sum_{i=1}^{n} p_{ik}^2}. \tag{5}$$

For categorical data the dissimilarity measure is

$$\mathbb{1}_{(x_{im}, c_{km})} = \begin{cases} 0 & x_{im} = c_{km} \\ 1 & x_{im} \neq c_{km}, \end{cases}$$

the JDF cannot be minimized through direct differentiation. The minimum is obtained when the cluster center c_{km} are equal to the highest number of observations since when $\mathbb{1}_{(x_{im}, c_{km})} = 0$, then, $\frac{\mathbb{1}_{(x_{im}, c_{km})} p_{ij}^2}{q_j} = 0$. It follows that the mode of the variable $\mathbf{x}_m$, for categorical variables, will minimize the JDF, i.e.,

$$c_{km} = \max_{s_m} \sum_{i=1, x_{im} = s_m}^{n} x_{im}^k, \tag{6}$$

where x_{im}^k means that x_{im} belongs to cluster k and s_m are the modalities of the variable m.

3.2. Algorithm

Cluster center initialization is obtained using partition around medoids (PAM) [28]. The dissimilarity between each point and each center is computed using (4), $\mathbf{D}$ is the matrix of dissimilarities of elements D_{ik}, and the probabilities are updated using (1). Given the new dissimilarities and probabilities the parameters can be updated using (2) for q_k and (3), (5), or (6) for the centers according to the type of variable. The difference between the previous and new centers is calculated. As proposed in [24] the algorithm convergences when the difference among the centers in two successive iterations is smaller than a predetermined threshold or the algorithm reaches an established maximum number of iterations. The matrix of probabilities $\mathbf{P}$ of general elements p_{ik} and the centers $\mathbf{c}_k$ are the output. The matrix $\mathbf{P}$ can be used to obtain a crisp classification assigning each point to the cluster corresponding the biggest value of p_{ik}. The algorithm steps are summarized in the flow chart in Fig. 1.

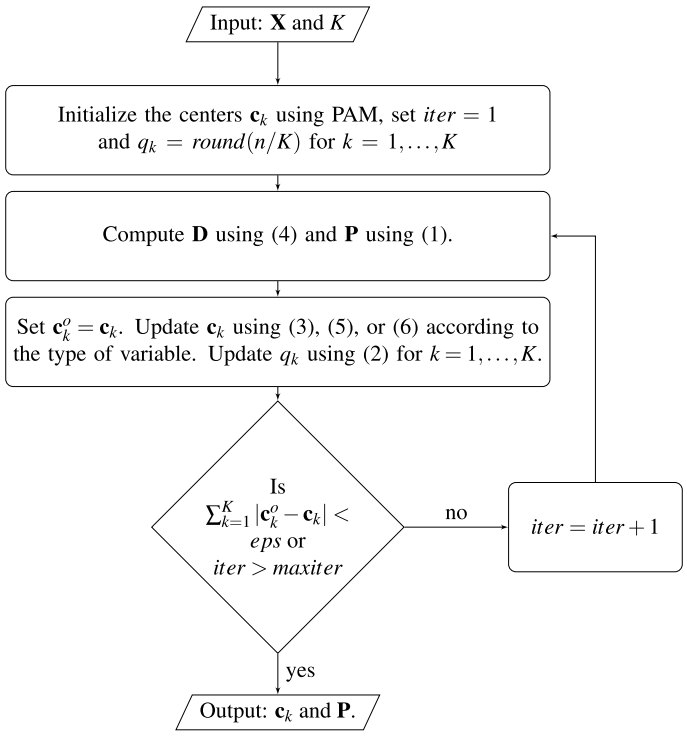

Fig. 1. Flow chart of the PDQ algorithm for mixed-type data.

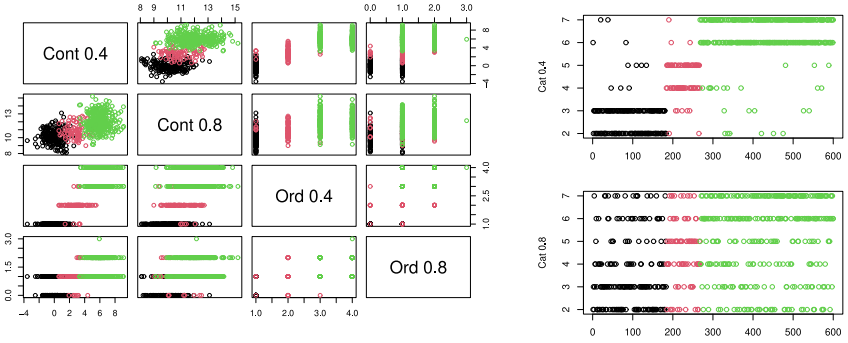

Fig. 2. Example of a simulated data set with 3 clusters, 2 continuous, 2 ordinal, and 2 categorical variables. One variable per kind has 0.4 and 0.8 overlap. The color and shape represent the cluster partition.

4. Application

4.1. Mixed-type data clustering algorithms

In the following sections, PDQ is used on real and simulated data. On the same data, K-prototypes [19] and Kamila (KAy-means for MIxed LArge data sets) [20] are used. Other techniques for mixed-type data exist, including many two-steps approaches, however the selected competitors have shown better results [29]. Both methods rely on one of the most widely used distance-based clustering algorithms: the K-means algorithm [30]. K-means goal is to minimize the within-cluster sum of the squared distances. It is achieved by randomly initializing the cluster centers at first, then the distance between each data point and the cluster centers is measured using the Euclidean distance to find the cluster membership. Finally, the centers are updated as cluster mean, and the algorithm iterates until convergence. The algorithm has been extended to cluster categorical data using simple matching as distance and mode as centers, namely the K-modes algorithm [31]. K-prototypes uses a dissimilarity measure between two observations which sums the mismatches for the categorical variables and the sum of the squared Euclidean distance for continuous variables into one measurement of dissimilarity, using a weight. The optimal centers are calculated individually for the continuous and categorical variables as the weighted mean and mode, respectively. Kamila has a semiparametric approach. The categorical variables are modeled as multivariate multinomial distributions, with dimensions equivalent to the number of discrete variables in the data (e.g., each observation of a data set with three categorical variables is modeled as a multivariate multinomial observation of three random variables). The continuous variables are modeled as multivariate random variables where a univariate probability density function is estimated through the kernel density transformation method. Each observation is classified into the cluster that maximizes the following qualifier:

$$H_i^{(t)}(k) = log[\hat{f}_v^{(t)}(d_{ik})^{(t)}] + log[c_{ik}^{(t)}],$$

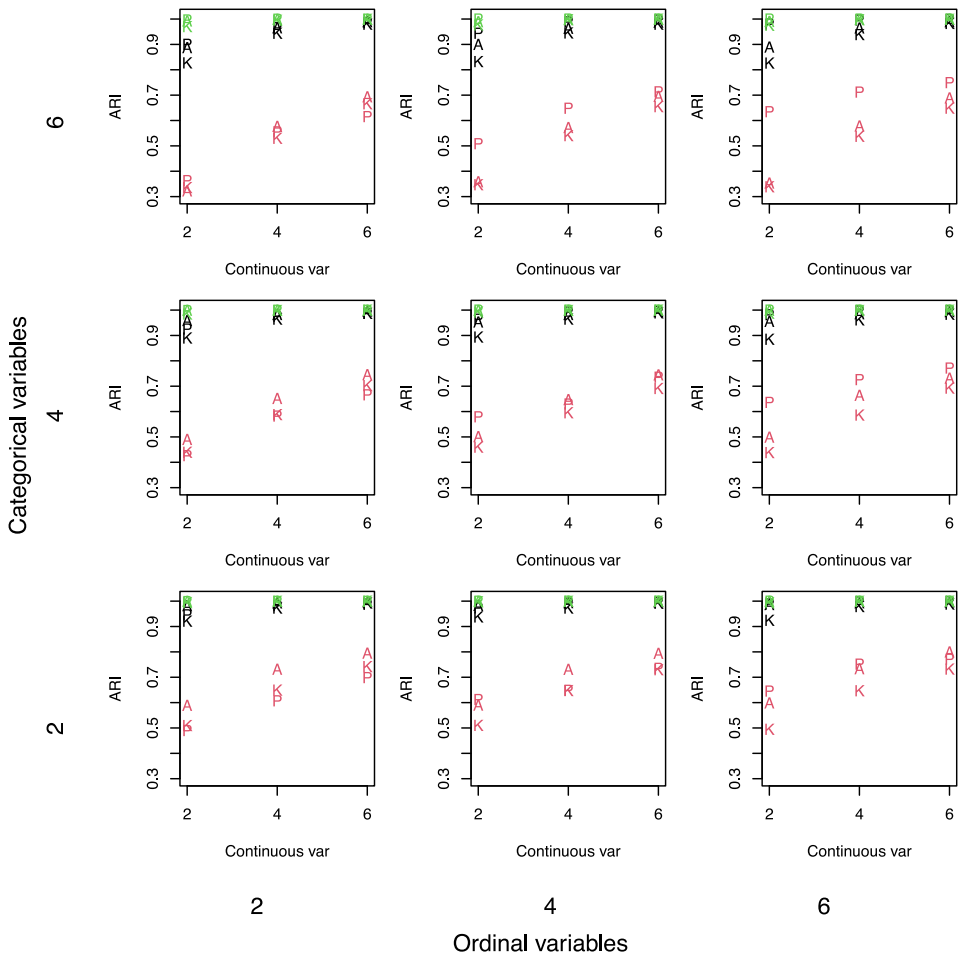

Fig. 3. Simulation results for 3 Clusters, average ARI varying the number of continuous, categorical, and ordinal variables. 'P' stands for PDQ, 'A' for Kamila, and 'K' for Kprototypes. Black represents 0.6 overlap with no correlation, red 0.8 overlap no correlation, green 0.4 overlap with random correlation.

where $k = 1, \ldots, K$ is the cluster, $i = 1, \ldots, n$ is the observation, t is the algorithm iteration, $\hat{f}_v^{(t)}(d_{ik})^{(t)}$ is the univariate kernel density estimate of the ith observation evaluated at $d_{ik}^{(t)}$, $d_{ik}^{(t)}$ is the distance between observation i and the center μ_k for cluster k, and $c_{ik}^{(t)}$ is the kth cluster multivariate multinomial probability for observation i.

Unfortunately, both K-prototype and Kamila algorithms are limited to categorical and continuous variables and do not deal with ordinal variables.

4.2. Simulation design

The analysis was done in R [32], both competitor algorithms are available on Cran R, the package `clustMixType` [19] implements K-prototypes, while `kamila` [20] implements Kamila. Since K-prototypes and Kamila only work on continuous or categorical variables, ordinal variables have been treated as categorical. Both algorithms use random starts, and we used five random starts. The code for the PDQ with mix-type data is available in the package `FPDclustering` [33]. To measure the quality of a partition, we used the adjusted Rand index (ARI). The ARI corrects the Rand index [34] for chance; it has an expected value equal to zero under random classification and is equal to one when there is a perfect class agreement. The ARI can be obtained using the `ARI` function of the `MixGHD` package [35]

Data were simulated using a similar methodology to the one used by McParland and Gormley 2016 [16]. Continuous data were generated from a multivariate Gaussian density with the dimension corresponding to the number of continuous variables desired. The means of the clusters set the levels of overlap. For continuous variables, the means for cluster 1 were obtained by equally partitioning the interval [0, 10] into the number of variables to simulate. For example, if three continuous variables were simulated, then the starting means for cluster 1 would be: (0, 5, 10). The means for the observations belonging to the other clusters were computed according to the desired overlap: $\mu_{k+1} = \mu_k + 5 - (ovlp * 5)$. For example a 10% overlap would correspond with the following means for 3 clusters: (0, 5, 10), (4.5, 9.5, 14.5), (9, 14, 19). For categorical variables, we assume that the mth variable with G_m possible responses has an underlying continuous vector that has $G_m - 1$ dimensions, i.e., $\mathbf{z}_{im} = \left(z_{im}^1, \ldots, z_{im}^{G_m-1}\right) \sim \text{MVN}_{G_m-1}\left(\boldsymbol{\mu}_m, \Sigma_m\right)$, where *MVN* denotes the multivariate Gaussian distribution. The observed categorical response y_{im} is a manifestation of the values of the elements of $\mathbf{z}_{ij}$ relative to each other and to a threshold, assumed to be 0 . That is,

$$x_{im} = \begin{cases} 1 \text{ if } \max_s \left\{z_{im}^s\right\} < 0 \\ g \text{ if } z_{im}^{g-1} = \max_s \left\{z_{im}^s\right\} \text{ and } z_{im}^{g-1} > 0 \text{ for } s = 2, \ldots, G_m. \end{cases} \tag{7}$$

Therefore a multivariate Gaussian distribution is used to simulate the $\mathbf{z}_{ij}$. We choose $2 * K$ dimension, corresponding to the modalities. The vector $\boldsymbol{\mu}_1$ is 2.5 for dimensions 1 and 2, and $2.5 * ovlp$ for the remaining dimensions. For cluster 2 $\boldsymbol{\mu}_2$, we shift the 2.5 to the

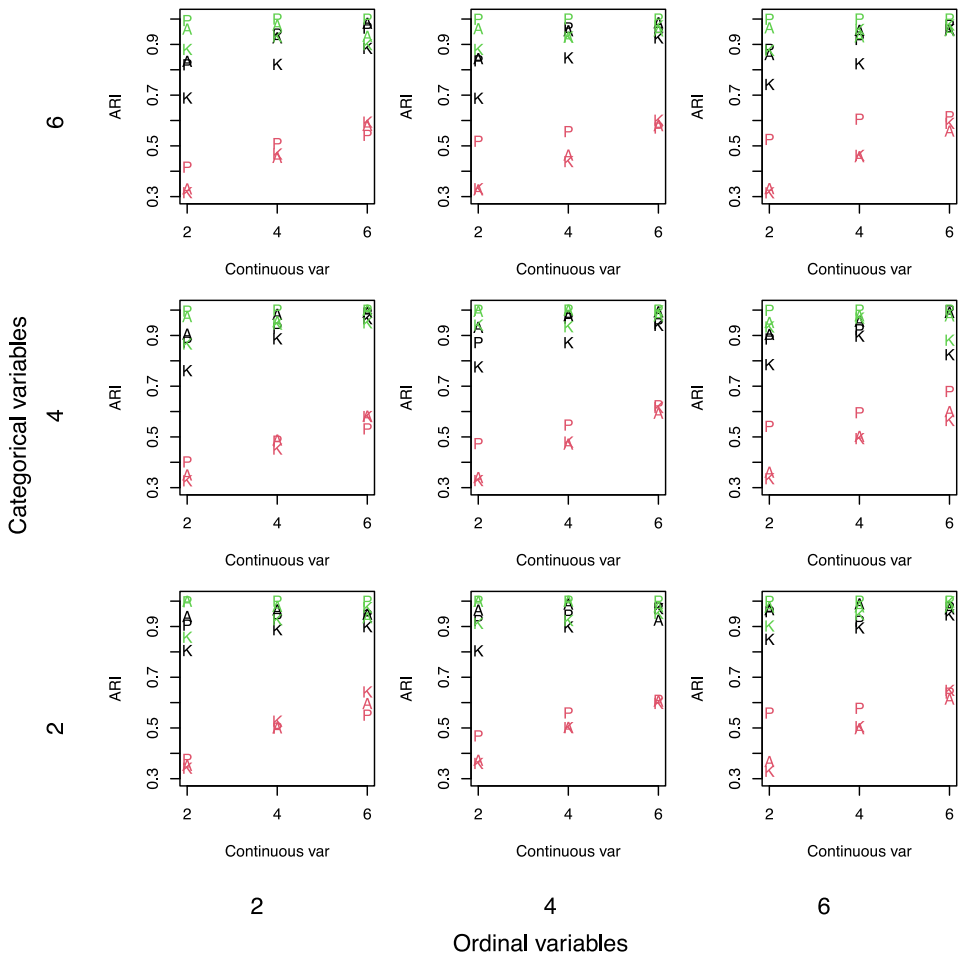

Fig. 4. Simulation results for 6 Clusters, average ARI varying the number of continuous, categorical, and ordinal variables. 'P' stands for PDQ, 'A' for Kamila, and 'K' for Kprototypes. Black represents 0.6 overlap with no correlation, red 0.8 overlap no correlation, green 0.4 overlap with random correlation.

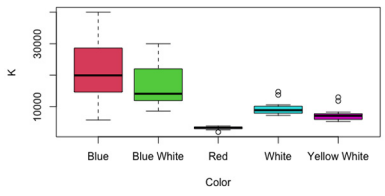

Fig. 5. Box plot of the variable Absolute temperature in K per color.

dimensions 3 and 4, and the remaining dimensions are calculated as $2.5{*}ovlp$, and so on. The categorical variables are then obtained using Eq. (7).

Similarly to categorical data, ordinal data, with O levels, is simulated through the assumption that it follows a latent variable $z \sim N(\mu, \sigma^2)$. Using a partitioned interval such that: $-\infty = \alpha_1 < \alpha_2 < \cdots < \alpha_O = \infty$.

Classification occurs with the following criteria:

$$\alpha_{o-1} < z < \alpha_o,$$

then the observed value $x_i = o$.

Ordinal data were simulated using the same method as the continuous variables, with the mean for cluster 1 was initialized by partitioning the interval [0, 10], where the number of partitions is equivalent to the number of simulated ordinal variables. The means are updated for each cluster using the following formula $\mu_{o+1} = \mu_o + (7 - 7 * ovlp)$. All the variance covariance matrices Σ_m were either diagonal or randomly generated using the R function `genPositiveDefMat` from the package `clusterGeneration` [36]. Fig. 2 shows an example of a simulated data set with 3 clusters, 2 continuous, 2 ordinal, and 2 categorical variables, with no correlation. One variable per kind has 0.4 and 0.8 overlaps. The color and shape represent the cluster partition.

4.3. Simulation results

For the simulation, we fixed the number of continuous, categorical, and ordinal variables equal to 2, 4, or 6, and we considered all the possible combinations. We fixed the number of observations equal to 600 and balanced it among clusters. We then varied the overlap between 0.6 and 0.8 with no correlation among variables and 0.4 with random correlation. Results are shown in Fig. 3. We repeated the same simulations with 6 clusters, and the results are in Fig. 4. We generate 10 data sets for each scenario, and the plots show the average ARI. The letters indicate the method in the plots: 'P' for PDQ, 'A' for Kamila, and 'K' for Kprototypes. The colors represent the level of overlapping, black for 0.6 with no correlation, red for 0.8 no correlation, and green for 0.4 with random correlation. In the data sets with 3 clusters, Fig. 3, all the techniques have similar performances with a lower overlap, 0.6, and no correlation (black). The overlap has the most significant impact on the ARI: when the overlap is 0.8 (red), the ARI decreases for all the methods and scenarios. PDQ

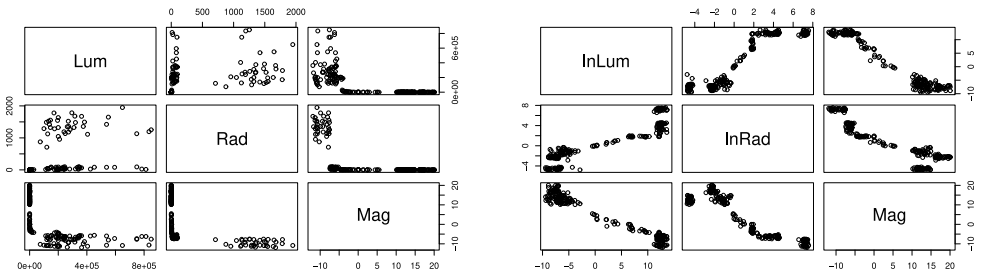

Fig. 6. Scatter plot of the variables relative luminosity, relative radius, and absolute magnitude on the left and of log relative luminosity, log relative radius, and absolute magnitude on the right.

Fig. 7. Silhouette plot of the partition in 6 clusters of the Star data set using PDQ.

Table 1
Average ARI of cluster partition vs. type.

	PDQ	Kamila	Kprototypes
ARI	0.8449	0.6290	0.6048

Table 2
Confusion matrix of the cluster partition obtained using PDQ vs. type.

Cluster	Red D.	Brown D.	White D.	Main Seq.	Super G.	Hyper G.
5	39	4	0	0	0	0
6	1	36	0	6	0	0
2	0	0	40	0	0	0
4	0	0	0	24	0	0
1	0	0	0	6	40	0
3	0	0	0	0	0	37

performs better when there are more categorical and/or ordinal variables. Kprototypes algorithm has the lowest ARI in general.

With 6 clusters, Fig. 4, Kprototypes performs worst than the other two techniques almost for all scenarios. PDQ and Kamila have similar behaviors, although PDQ performs better with correlated variables.

4.4. Star classification

The Star data set is available on Kaggle,[1] the goal is to recognize the start type given some characteristics. The available variables are:

- Absolute Temperature (in K, continuous)
- Relative Luminosity (L/Lo, continuous)
- Relative Radius (R/Ro, continuous)
- Absolute Magnitude (Mv,continuous)
- Star Color (white,Red,Blue,Yellow,yellow–white, blue–white)
- Spectral Class (O,B,A,F,G,K,M)

The star types are: Red Dwarf, Brown Dwarf, White Dwarf, Main Sequence, SuperGiants, HyperGiants.

Fig. 5 shows the box plot of the variable Absolute temperature (symbol K on the vertical axis) by the color; red, white, and yellow white stars appear clearly distinct, blue and white blue have high variability in temperature. Both color and temperature depend on the star radiate energy; therefore, keeping both did not make sense, and we dropped the absolute temperature variable.

Moreover, looking at Fig. 6 we decide to use a logarithmic transformation for the variables, Relative Luminosity, and Relative Radius. We set the number of clusters equal to 6 for all the methods.

Table 1 shows the ARI for the three methods, PDQ performs better than the other two with a big margin; the PDQ ARI is 0.85 versus 0.63 and 0.60 for Kamila and Kprototypes, respectively (average). Kamila and Kprototypes had high variability in the results; the table shows the average ARI obtained on 25 iterations the corresponding standard deviations are 0.1066 for Kamila and 0.1152 for Kprototypes.

Table 2 shows the confusion matrix for the PDQ. Most clusters are very well separated, except the type Main Sequence, which gets mixed up with Super Giants and Brown Dwarf. Fig. 7 shows a probabilistic Silhouette plot. All the clusters are very well separated with high belonging probabilities, clusters 6 and 4 corresponding to Brown Dwarf, and Main Sequence have some points with lower belonging probabilities. Those are two of the clusters that contain miss-classified points. Cluster 1, Super Giants, also contains some miss-classified points, but this is not obvious in the silhouette plot.

Fig. 8 shows a scatter plot and a parallel coordinate plot of the continuous variables of the Star data set. We can see that clusters 3 Hyper Giants, 1 Super Giant, and 4 Main Sequence are well separated on the continuous variables. Brown Dwarf, cluster 6, slightly overlaps with Red and white Dwarf, clusters 5 and 2 respectively. Looking at the categorical variables Fig. 9, cluster 5 Red Dwarfs, and Brown Dwarfs, cluster 6 are very homogeneous and clearly different form White Dwarf, cluster 2, and 4 Super Giant. The categorical variables also separate clusters 1 and 4 corresponding to Main Sequence and Super Giants. Those plots clearly emphasize how both groups of variables, continuous and categorical, contribute to the correct partition of the stars.

4.5. Other fields of application

The Star classification problem is a suitable example to appreciate the PDQ clustering potentialities, showing the PDQ flexibility in treating different types of variables. PDQ does not require strong distributional assumptions on the data and can be applied in several fields: in medicine or psychology, for example, where patients' records contain data of mixed types, and clustering is a tool that helps the diagnostics. Mixed data generally arise also when dealing with customer profiling problems. Clustering is also

[1] https://www.kaggle.com/deepu1109/star-dataset.

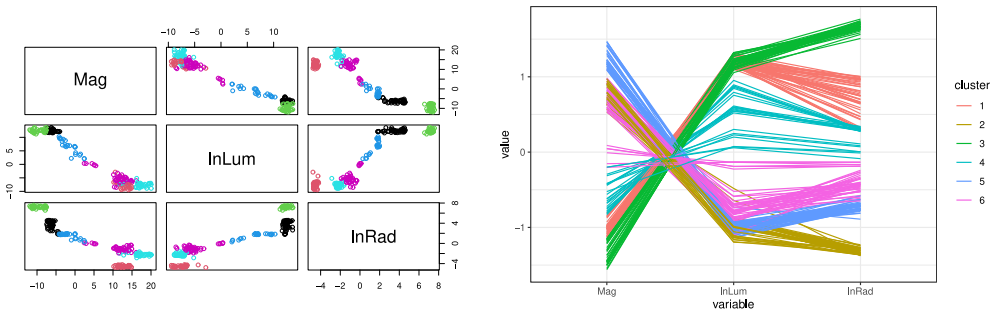

Fig. 8. Scatter plot and parallel plot of the continuous variables of the star data set. The colors represent the partition obtained with the PDQ.

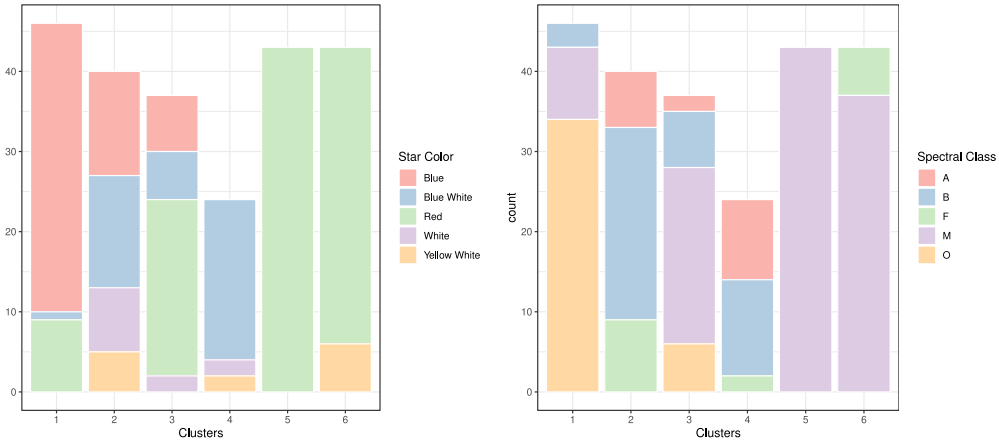

Fig. 9. Bar plot of the categorical variables of the star data set. The colors represent the categorical variables and the x axes the partition obtained with the PDQ.

a step in more complex models or analyses. For example, in process control, a different equation-type model can be used in each cluster to describe the input–output-relations within the cluster [37]. Recently, many fields historically unrelated to clustering have started using cluster analysis models: graph theory and networks analysis exploit cluster analysis capabilities to afford several categorization problems in many domains. Clusters analysis also helps to increase the lifetime of various wireless sensor networks efficiently [38]. The contribute [39] offers many other examples in several domains of cluster analysis applications.

5. Conclusion

Real data analysis increasingly involves variables of mixed-type, i.e., continuous, ordinal, and categorical, with a consequent increase in the need for clustering algorithms capable of finding clusters, i.e., homogeneous groups of units within the data when the variables are mixed-type. This work extends probabilistic distance clustering adjusted for cluster size (PDQ) for this purpose. PDQ associates a fuzzy cluster membership to each observation and has shown promising results on simulated and real data.

PDQ has the advantage of overcoming several limitations of clustering methods for mixed-type data based on a geometric approach. One of them is that the weighting considers the inner cluster variability. However, it cannot account for the correlation among the variables (see also [14]). Nevertheless, considering correlations when dealing with mixed-type variables turns out quite challenging, and conditional independence assumption is common even in a mixture model context, where correlation is usually estimated [40]. Future research can go in several different directions; research into different weight measures for the different types of variables is limited, which leaves room for further exploratory studies, together with an extension of factor probabilistic distance clustering (FPDC), specific for high dimensional data, to mixed-type data. Another opportunity for future exploratory studies is the choice of the algorithm, the current algorithm, although pretty fast, can still converge to local optima. A possible solution is to consider MCMC-like algorithms [41].

The software used for this paper is available on CRAN, R package `FPDclustering` [33]. The code for mixed-type data clustering is part of the `PDQ` function, specifying the appropriate distance.

CRediT authorship contribution statement

Cristina Tortora: Conceptualization, Methodology, Software, Formal analysis, Writing, Funding acquisition. **Francesco Palumbo:** Conceptualization, Methodology, Software, Formal analysis, Writing.

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.

Data availability

The data are available on R

Acknowledgments

We want to thank Noe Vidales, former Master student at San José State University, for his contribution to this work during his master thesis.

Funding

Research reported in this publication was supported by Woodward funds of San José State University Mathematics and Statistics department award number 3415040090 and by the Central RSCA of San José State University award number 18-RSG-08-046.

References

[1] A.D. Gordon, Classification, second ed., Chapman and Hall/CRC, Boca Raton, 1999.
[2] S.K. Ng, T. Krishnan, G.J. McLachlan, The EM algorithm, in: J. Gentle, et al. (Eds.), Handbook of Computational Statistics, Springer Verlag (Germany), Berlin, Heidelberg, 2012, pp. 139–172.
[3] G. Celeux, G. Govaert, A classification EM algorithm for clustering and two stochastic versions, Comput. Statist. Data Anal. 14 (3) (1992) 315–332.
[4] G.J. McLachlan, D. Peel, Finite Mixture Models, Wiley Interscience, New York, 2000.
[5] G.J. McLachlan, T. Krishnan, The EM Algorithm and Extensions, Wiley interscience, 2008.
[6] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, New York, NY, 2006.
[7] A. Ben-Israel, C. Iyigun, Probabilistic D-clustering, J. Classification 25 (1) (2008) 5–26.
[8] C. Iyigun, Probabilistic Distance Clustering (Ph.D. thesis), New Brunswick Rutgers, The State University of New Jersey, 2007.
[9] C. Tortora, Non-Hierarchical Clustering Methods on Factorial Subspaces (Ph.D. thesis), Università di Napoli Federico II, 2011.
[10] C. Tortora, M. Gettler Summa, F. Palumbo, Factor PD-clustering, in: B. Lausen, D. Van den Poel, A. Ultsch (Eds.), Algorithms from and for Nature and Life, 2013, pp. 115–123.
[11] L.R. Tucker, Some mathematical notes on three-mode factor analysis, Psychometrika 31 (3) (1966) 279–311.
[12] P.M. Kroonenberg, Applied Multiway Data Analysis, Ebooks Corporation, Hoboken, New Jersey, 2008.
[13] C. Tortora, M. Gettler Summa, M. Marino, F. Palumbo, Factor probabilistic distance clustering (FPDC): A new clustering method for high dimensional data sets, Adv. Data Anal. Classif. 10 (4) (2016) 441–464.
[14] C. Tortora, P.D. McNicholas, F. Palumbo, A probabilistic distance clustering algorithm using Gaussian and student-t multivariate density distributions, SN Comput. Sci. 1 (2) (2020) 1–22.
[15] M. Ichino, H. Yaguchi, General Minkowski metrics for mixed features type data analysis, IEEE Trans. Syst. Man Cybern. 24 (1994) 698–708.
[16] D. McParland, I.C. Gormley, Model based clustering for mixed data: clustMD, Adv. Data Anal. Classif. 10 (2016) 155–169.
[17] F. Mbuga, C. Tortora, Spectral clustering of mixed-type data, Stats 5 (1) (2022) 1–11.
[18] C. Li, G. Biswas, Unsupervised learning with mixed numeric and nominal data, IEEE Trans. Knowl. Data Eng. 14 (4) (2002) 673–690.
[19] G. Szepannek, ClustMixType: User-friendly clustering of mixed-type data in R, R J. 10 (2) (2018) 200–208.
[20] A.H. Foss, M. Markatou, Kamila: Clustering mixed-type data in R and Hadoop, J. Stat. Softw. 83 (13) (2018) 1–45.
[21] L. Hunt, M. Jorgensen, Clustering mixed data, Wiley Interdiscip. Rev.: Data Min. and Knowl. Discov. 1 (4) (2011) 352–361.
[22] A. Ahmad, S.S. Khan, Survey of state-of-the-art mixed data clustering algorithms, Ieee Access 7 (2019) 31883–31902.
[23] M. van de Velden, A. Iodice D'Enza, A. Markos, Distance-based clustering of mixed data, Wiley Interdiscip. Rev. Comput. Stat. 11 (3) (2019) e1456.
[24] C. Iyigun, A. Ben-Israel, Probabilistic distance clustering adjusted for cluster size, Probab. Engrg. Inform. Sci. 22 (04) (2008) 603–621.
[25] C. Rainey, C. Tortora, F. Palumbo, A parametric version of probabilistic distance clustering, in: Scientific Meeting of the Classification and Data Analysis Group of the Italian Statistical Society, Springer, 2017, pp. 33–43.
[26] J.C. Gower, A general coefficient of similarity and some of its properties, Biometrics 27 (4) (1971) 857–871.
[27] G.W. Milligan, M.C. Cooper, A study of standardization of variables in cluster analysis, J. Classification 5 (2) (1988) 181–204.
[28] L. Kaufman, P. Rousseeuw, Y. Dodge, Clustering by means of medoids in statistical data analysis based on the ℓ_1 norm and related methods, 1987, pp. 405–416.
[29] J. Jimeno, M. Roy, C. Tortora, Clustering mixed-type data: A benchmark study on KAMILA and K-prototypes, in: Data Analysis and Rationality in a Complex World 16, Springer International Publishing, 2021, pp. 83–91.
[30] J. MacQueen, Some methods for classification and anlysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium, Vol. 1, 1967, pp. 281–297.
[31] A. Chaturvedi, P.E. Green, J.D. Caroll, K-modes clustering, J. Classification 18 (1) (2001) 35–55.
[32] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2016.
[33] C. Tortora, N. Vidales, F. Palumbo, T. Kalra, P.D. McNicholas, FPDclustering: PD-clustering and factor PD-clustering, 2022, R Package Version 2.1.
[34] W.M. Rand, Objective criteria for the evaluation of clustering methods, J. Amer. Statist. Assoc. 66 (1971) 846–850.
[35] C. Tortora, R.P. Browne, A. ElSherbiny, B.C. Franczak, P.D. McNicholas, Model-based clustering, classification, and discriminant analysis using the generalized hyperbolic distribution: MixGHD R package, J. Stat. Softw. 98 (1) (2021) 1–24.
[36] W. Qiu, H. Joe, Qiu, The clustergeneration package, 2006, version 1.3.7.
[37] R. Ylinen, S.-L. Jämsä-Jounela, J. Miettunen, Use of cluster analysis in process control, IFAC Proc. Vol. 26 (2) (1993) 645–648.
[38] X. Zhao, Z. Gao, R. Huang, Z. Wang, T. Wang, A fault detection algorithm based on cluster analysis in wireless sensor networks, in: 2011 Seventh International Conference on Mobile Ad-Hoc and Sensor Networks, IEEE, 2011, pp. 354–355.
[39] C.C. Aggarwal, An introduction to cluster analysis, in: Data Clustering, Chapman and Hall/CRC, 2018, pp. 1–28.
[40] R.P. Browne, P.D. McNicholas, Model-based clustering, classification, and discriminant analysis of data with mixed-type, J. Statist. Plann. Inference 142 (11) (2012) 2976–2984.
[41] L. Martino, V. Elvira, Metropolis sampling, 2017, arXiv preprint arXiv: 1704.04629.