Journal of Classification (2023) 40:2–24
https://doi.org/10.1007/s00357-022-09422-y

Imputation Strategies for Clustering Mixed-Type Data with Missing Values

Rabea Aschenbruck[1] · **Gero Szepannek**[1] · **Adalbert F. X. Wilhelm**[2]

Accepted: 12 September 2022 / Published online: 26 November 2022

Abstract
Incomplete data sets with different data types are difficult to handle, but regularly to be found in practical clustering tasks. Therefore in this paper, two procedures for clustering mixed-type data with missing values are derived and analyzed in a simulation study with respect to the factors of partition, prototypes, imputed values, and cluster assignment. Both approaches are based on the k-prototypes algorithm (an extension of k-means), which is one of the most common clustering methods for mixed-type data (i.e., numerical and categorical variables). For k-means clustering of incomplete data, the k-POD algorithm recently has been proposed, which imputes the missings with values of the associated cluster center. We derive an adaptation of the latter and additionally present a cluster aggregation strategy after multiple imputation. It turns out that even a simplified and time-saving variant of the presented method can compete with multiple imputation and subsequent pooling.

Keywords Clustering · Imputation · Mixed-type data · Missing values

1 Introduction

Cluster analysis is a technique for discovering structural similarity in data, with the particular goal of identifying unknown groups in the data (Hennig et al., 2015). One of the most popular and widely used cluster techniques is the k-means algorithm, which has experienced further developments in the past (for more information, see, e.g., Jain, 2010). Since it has become the standard in practice to cluster mixed-type data, various approaches were presented in the past (Ahmad & Khan, 2019) and numerous articles have been published on applications in a variety of disciplines (see, for example, van 't Veer et al., 2002; Hennig

✉ Rabea Aschenbruck
rabea.aschenbruck@hochschule-stralsund.de

Gero Szepannek
gero.szepannek@hochschule-stralsund.de

Adalbert F. X. Wilhelm
a.wilhelm@jacobs-university.de

[1] Hochschule Stralsund - University of Applied Sciences, Zur Schwedenschanze 15, 18435 Stralsund, Germany

[2] Jacobs University Bremen, Campus Ring 1, 28759 Bremen, Germany

& Liao, 2013; Yin et al., 2021). For this data, consisting of numerical and categorical variables, one of the most popular approaches for clustering based on k-means is the k-prototypes algorithm proposed by Huang (1998).

In practice it is quite common to work with incomplete data. However, handling these missing values is usually not trivial. The simplest and most straightforward way to deal with missing values regardless of the examination method is complete case analysis, i.e., to delete all incomplete observations (Little & Rubin, 2019). The following aspects are to be considered: The partly known information of the incomplete observations is not included in the analysis result and the results might be biased. Furthermore, in the cluster analysis no cluster assignment can be provided to the deleted observations. In 2004, Wagstaff investigated whether cluster analysis can be performed for all observations without imputation. However, the majority of approaches for dealing with missing values in cluster analysis is based on (multiple) imputation and cluster aggregation (Basagaña et al., 2013; Audigier & Niang, 2020). In addition to the imputation of the values (Imbert & Vialaneix, 2018; Little & Rubin, 2019), the aggregation method of the partitions is of particular importance and different approaches were examined, for example, in Gionis et al. (2005).

While dealing with missing data is standard practice when applying the k-means algorithm to purely numeric data, handling incomplete mixed-type data is often challenging. Some approaches on imputing missing values in the application of k-prototypes have been developed, but are not satisfying with respect to imputation and/or taking the mixed-type structure into account (e.g., imputation in Dinh et al., 2021, is sequentially performed based only on the similarity of the categorical values). Recently, for standard k-means clustering of incomplete numerical data, the k-POD algorithm has been proposed by Chi et al. (2016). During the clustering task, the missing entries are estimated from the corresponding entries of the relevant cluster centroid. The adaptation of this procedure to cluster analysis of mixed-type data is derived in this paper. Additionally, a cluster aggregation strategy after multiple imputation based on Gionis et al. (2005) is adapted for mixed-type data. These different approaches are analyzed comparatively in a simulation study.

The rest of the paper is organized as follows: First, the theory of handling missing values and the adaptation of the k-POD idea is explained (Section 2) as well as the usage of multiple imputation of chained equations and pooling of the cluster results for imputation (Section 3). In Section 4, a simulation study on synthetic data is conducted and shows the potential of the examined methods with respect to the criteria of partition quality, the specified prototypes, the imputed values, and also the cluster assignment. All aspects will be evaluated in comparison to the original complete data, made possible by artificially added missing values in the data.

2 Applying k-Prototypes Algorithm to Achieve Partition and Imputation of Missing Values

In the following, the adaptation of k-POD for numerical data (Chi et al., 2016) to k-prototypes for mixed-type data is outlined. Before going into more detail about the method for mixed-type data, basic notations for representing the data and the k-prototypes algorithm are presented. Subsequently in Section 2.2, the minimization problem of the cluster algorithm for mixed-type data and its solution are shown in Theorem 1.

Finally, variations of imputation using the k-prototypes algorithm are differentiated in Section 2.3.

2.1 Notation

Let $X \in \mathbb{M}^{n\times m}$ denote a matrix of n observations of m variables of mixed type, with

$$\mathbb{M}^{n\times m} := \big\{[X_i]_{i=1,\dots,n} = [(x_{i1},\dots,x_{il},x_{i(l+1)},\dots,x_{im})]_{i=1,\dots,n} \mid x_{ij} \in \mathbb{R}\ \forall j \in \{1,\dots,l\} \wedge x_{ij} \in A_j\ \forall j \in \{l+1,\dots,m\}\big\},$$

where $\wedge$ denotes a *logical and*, so there is $x_{i1},\dots,x_{il}$ numerical data and $x_{i(l+1)},\dots,x_{im}$ categorical data without loss of generality (w.l.o.g.), where A_j describes a domain of values of variable j.

After applying the k-prototypes cluster algorithm to the data, the k cluster prototypes will be denoted by $C \in \mathbb{M}^{k\times m}$, where $c_q \in \mathbb{M}^{1\times m}$ with $q \in \{1,\dots,k\}$ denotes the prototype of cluster q. In the k-prototypes algorithm, the distances between mixed-type objects result from the sum of the Euclidean distance for the numerical features and the so-called *Simple Matching* indicator function

$$\delta(x_{aj}, x_{bj}) = \begin{cases} 0, & x_{aj} = x_{bj} \\ 1, & x_{aj} \neq x_{bj} \end{cases}$$

for the categorical features. It is defined as

$$d_{MT}(X_a, X_b) = \sum_{j=1}^{l}(x_{aj} - x_{bj})^2 + \lambda \sum_{j=l+1}^{m} \delta(x_{aj}, x_{bj}) \tag{1}$$

with $X_a, X_b \in \mathbb{M}^{1\times m}$ and weighting factor $\lambda > 0$ (for more details, see Huang, 1998). The k-prototypes algorithm minimizes the sum of the distances between the clustered objects and their respective cluster prototypes

$$\sum_{i=1}^{n} d_{MT}(X_i, K_i(C, W)). \tag{2}$$

The notation $K_i(C,W)$ denotes the associated prototype of the cluster of X_i and is defined as

$$K(C, W) := [K_i(C, W)]_{i=1,\dots,n} = [K_i(C, W) = c_q \mid w_{iq} = 1]_{i=1,\dots,n} \in \mathbb{M}^{n\times m},$$

where w_{iq} is an element of the partition matrix $W \in H$. For example, $w_{i3} = 1$ denotes that object i is assigned to cluster 3, and therefore $w_{iq} = 0 \forall q \neq 3$ due to the fact that object i can only be assigned to exactly one cluster (since this is not fuzzy clustering, obviously). The partition matrix W is element of the set H, which is defined as

$$H = \big\{W \in \mathbb{R}^{n\times k} \mid W\mathbf{1_k} = \mathbf{1_n} \wedge w_{iq} \in \{0,1\}\ \forall i \in \{1,\dots,n\}, q \in \{1,\dots,k\}\big\}.$$

In the paper at hand, the notation for the representation of observed and missing data is based on a set $\Omega \subseteq \{1,\dots,n\} \times \{1,\dots,m\}$. The indices of the observed data are stored in Ω and the projection $P_\Omega(X_i)$ contains only the known data without the unobserved values:

$$[P_{\Omega}(X_i)]_j = \begin{cases} x_{ij}, & (i,j) \in \Omega, \\ 0, & (i,j) \notin \Omega \wedge j \in \{1,\dots,l\}, \\ "NA", & (i,j) \notin \Omega \wedge j \in \{l+1,\dots,m\}. \end{cases}$$

Assuming that the value x_{ij} is not known, then $(i,j) \notin \Omega$ holds. Depending on the nature of the variable j, it implies $[P_{\Omega}(X_i)]_j = 0$ (variable j is numeric and therefore w.l.o.g. $j \in \{1,\dots,l\}$) or $[P_{\Omega}(X_i)]_j = "NA"$ (variable j is categorical and therefore w.l.o.g. $j \in \{l+1,\dots,m\}$) for the missing value. On the other hand, projection $P_{\Omega^C}(X_i)$ (with Ω^C the complementary set of Ω) can be used to denote the true but unknown values of the missing data of observation i.

2.2 Minimization Problem of Cluster Algorithm with Missing Values

As shown previously in (2), the k-prototypes algorithm minimizes the distances between the fully observed data and the prototypes of the partition. This paper investigates the approach in the case of incomplete data. Then the clustering problem is modified to the minimization of the distances between the observed data values and the associated prototypes, denoted as $d_{\mathrm{MT}}(P_{\Omega}(X_1), P_{\Omega}(K_i(C,W)))$. This leads to the minimization problem of the function

$$f(C,W) = \sum_{i=1}^{n} d_{\mathrm{MT}}(P_{\Omega}(X_i), P_{\Omega}(K_i(C,W)))$$

for the partition, defined by cluster prototypes C and partition matrix W.

The remainder of the section aims to derive the minimization of distances between mixed-type data with missing values and prototypes of associated clusters. The proof of the solution of this minimization problem of clustering with missing values applies the majorization-minimization algorithm (MM algorithm; Lange, 2013). Briefly summarized, it says that if a function $f(u)$ is to be minimized, a function $g(u|u^{(t)})$ with the following properties must be identified:

$$i)\ \ g(u|u^{(t)}) \text{ majorizes } f(u) (\Rightarrow f(u) \leq g(u|u^{(t)})\ \forall u) \text{ and} \tag{3}$$

$$ii)\ \ g(u|u^{(t)}) \text{ is anchored at } u^{(t)} (\Rightarrow f(u^{(t)}) = g(u^{(t)}|u^{(t)})). \tag{4}$$

The MM algorithm iterates $u^{(t+1)} := \mathrm{argmin}_u g(u|u^{(t)})$ and it holds

$$f(u^{(t+1)}) \leq g(u^{(t+1)}|u^{(t)}) \leq g(u^{(t)}|u^{(t)}) = f(u^{(t)}).$$

Finally, it can be concluded for $t \to \infty$: $\min_u f(u) = f(u^{(t+1)})$.

In the following, the solution to minimize the distances between incomplete mixed-type data and cluster prototypes during the clustering task is presented.

Theorem 1 The minimum of the distances between incomplete data and prototypes of a clustering grouping this mixed-type data with missing values

$$f(C,W) = \sum_{i=1}^{n} d_{\mathrm{MT}}(P_{\Omega}(X_i), P_{\Omega}(K_i(C,W)))$$

can be found at $f(C^{(t+1)},W^{(t+1)})$ for $t \to \infty$, where

$$\left(C^{(t+1)}, W^{(t+1)}\right) := \underset{C,W}{\operatorname{argmin}} \sum_{i=1}^{n} d_{\mathrm{MT}}\left(\tilde{X}_i, K_i(C,W)\right)$$

and $\tilde{X} := \left[[P_{\Omega}(X_i)]_j \; \forall (i,j) \in \Omega \wedge [P_{\Omega^C}(K_i(C^{(t)}, W^{(t)}))]_j \; \forall (i,j) \notin \Omega\right]$, so that $\tilde{X}$ denotes the data, where the missing values are imputed by the respective prototype values.

The solution $(C^{(t+1)},W^{(t+1)})$ is obtained by the k-prototypes algorithm applied to $\tilde{X}$.

Proof The function $f(C,W) = \sum_{i=1}^{n} d_{\mathrm{MT}}\left(P_{\Omega}(X_i), P_{\Omega}(K_i(C,W))\right)$ is to be minimized. The majorization function g for the application of the MM algorithm is

$$g(C,W|C^{(t)},W^{(t)}) = f(C,W) + \sum_{i=1}^{n} d_{\mathrm{MT}}\left(P_{\Omega^C}(K_i(C,W)), P_{\Omega^C}(K_i(C^{(t)},W^{(t)}))\right), \tag{5}$$

and both conditions (3) and (4) of the MM algorithm are fulfilled:

i) Since $\sum_{i=1}^{n} d_{\mathrm{MT}}\left(P_{\Omega^C}(K_i(C,W)), P_{\Omega^C}(K_i(C^{(t)},W^{(t)}))\right)$ is the sum of distances for all (C,W), where each summand is greater than or equal to zero, (3) with $f(C,W) \le g(C,W|C^{(t)},W^{(t)})$ is given and g majorizes f.
ii) Furthermore, (4) is valid and g is anchored with

$$f(C^{(t)},W^{(t)}) = f(C^{(t)},W^{(t)}) + 0 = g(C^{(t)},W^{(t)}|C^{(t)},W^{(t)}).$$

Therefore, the minimum of f for $t \to \infty$ can be found at $f(C^{(t+1)},W^{(t+1)})$ with

$$\left(C^{(t+1)}, W^{(t+1)}\right) := \underset{C,W}{\operatorname{argmin}}\, g(C,W|C^{(t)},W^{(t)}).$$

It follows that

$$\begin{aligned}
g(C,W|C^{(t)},W^{(t)}) &= f(C,W) + \sum_{i=1}^{n} d_{MT}\left(P_{\Omega^C}(K_i(C,W)), P_{\Omega^C}(K_i(C^{(t)},W^{(t)}))\right) \\
&= \sum_{i=1}^{n} d_{MT}\left(P_{\Omega}(X_i), P_{\Omega}(K_i(C,W))\right) \\
&\quad + \sum_{i=1}^{n} d_{MT}\left(P_{\Omega^C}(K_i(C,W)), P_{\Omega^C}(K_i(C^{(t)},W^{(t)}))\right) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{l} \left([P_{\Omega}(X_i)]_j - [P_{\Omega}(K_i(C,W))]_j\right)^2 \\
&\qquad + \left([P_{\Omega^C}(K_i(C,W))]_j - [P_{\Omega^C}(K_i(C^{(t)},W^{(t)}))]_j\right)^2 \\
&\quad + \lambda \sum_{i=1}^{n}\sum_{j=l+1}^{m} \delta\left([P_{\Omega}(X_i)]_j, [P_{\Omega}(K_i(C,W))]_j\right) \\
&\qquad + \delta\left([P_{\Omega^C}(K_i(C,W))]_j, [P_{\Omega^C}(K_i(C^{(t)},W^{(t)}))]_j\right) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{l} \left([\tilde{X}_i]_j - [K_i(C,W)]_j\right)^2 + \lambda \sum_{i=1}^{n}\sum_{j=l+1}^{m} \delta\left([\tilde{X}_i]_j, [K_i(C,W)]_j\right) \\
&= \sum_{i=1}^{n} d_{MT}\left(\tilde{X}_i, K_i(C,W)\right),
\end{aligned} \tag{6}$$

where $\tilde{X} := \left[[P_{\Omega}(X_i)]_j \; \forall (i,j) \in \Omega \wedge [P_{\Omega^C}(K_i(C^{(t)}, W^{(t)}))]_j \; \forall (i,j) \notin \Omega\right]$.

The mathematical transformation for (6) is presented separately in the Appendix, for numerical values see Appendix A.1 and for categorical values see Appendix A.2.

The function $g(C,W|C^{(t)},W^{(t)})$ will be minimized with an application of the k-prototypes algorithm on the observed data filled with the prototype values $\tilde{X}$. With the resulting partition and associated cluster prototypes, $f(C,W) = \sum_{i=1}^{n} d_{\mathrm{MT}}\left(P_{\Omega}(X_i), P_{\Omega}(K_i(C,W))\right)$ is minimized and therefore the solution of the clustering problem for incomplete mixed-type data is obtained. □

In the Theorem 1 proved above, it is shown that by applying the k-prototypes algorithm, the minimization problem of the distances between the incomplete data and the prototypes of the partition can be solved. The implementation of the imputation used for this purpose is presented in the following.

2.3 Variations of Imputation with k-Prototypes Clustering

In the subsequent simulation study, three different variations of handling the missing values based on the approach presented above are investigated. The distinctions of the imputations based on the values of the prototypes are pointed out in the following:

i) *Internal all steps imputation*
   Within the k-prototypes algorithm, the missing values are imputed with the current prototype value after each assignment of new prototypes. This means that in each iteration, the closest prototype is determined for each observation (possibly with missing values) and the observation is assigned to this cluster. Then the prototypes are recalculated based on the observations in the respective cluster. The updated values of the prototypes are used to replace the missing values or the former imputations and with the new imputations the iteration starts over again. Finally, these iterations run until no further improvement of the partition can be observed or the number of iteration steps exceeds a predefined limit.
ii) *Repeated external imputation*
   As the name of the strategy implies, the imputation of the missing values by the values of the prototypes is performed after the entire application of the k-prototypes algorithm has terminated. Then, the cluster algorithm is run again with the imputed values and the (former) missing values of the data are updated with the new determined prototypes. This continues until there is no change in the imputed values or a predefined limit of iteration steps is reached.
iii) *Fast one step imputation*
   The last variation examined is a fast alternative to variant ii). The k-prototypes algorithm is applied only once, ignoring the missing values when determining the distances. Afterwards, the missing values are imputed with the corresponding values of the prototypes. The comparison in the simulation study between this variant and variant ii) will show whether the clearly higher computational costs of the latter one are justified by a distinctly better outcome.

The results of the analysis conducted in the simulation study with respect to different aspects are presented in Section 4.

3 Multiple Imputation and Pooling of k-Prototypes Cluster Results

Currently, one of the most popular approaches to handle missing values is multiple imputation. Another strategy for handling missing values using multiple imputation is therefore proposed in order to compare the quality of the adapted k-POD algorithm with common existing methods. Particular attention is paid to the way of cluster aggregation after the application of multiple imputation and cluster analysis in Section 3.2. First, a brief overview of multiple imputation for mixed-type data is provided.

3.1 Multiple Imputation by Chained Equations

The basic idea of multiple imputation is that the uncertainty resulting from the lack of knowledge of the missing values is expressed by multiple imputed values. With this strategy, n_{imp} data sets will be obtained with the same known values and (usually different) imputed values for the missing values. Thereby, a univariate imputation model is specified for each incomplete variable (in the following denoted by Y), which leads to the name *multiple imputation by chained equations*. The imputation result is based on the other variables, named $X = \{[X_j]_{j=1,\ldots,r}\}$. The specification is based on the scale of the variable with missing values.

The most common and straightforward imputation method for numerical data is the so-called *predictive mean matching* (Little, 1988). For each of the n_{imp} data sets this algorithm repeats the following steps:

i) Estimate a regression model for Y based only on the observed values $P_{\Omega}(X)$.
ii) Randomly draw from the posterior predictive distribution of the estimated regression parameter vector $\hat{\beta}$ and produce a new set of coefficients $\beta^{\star}$.
iii) Calculate predicted values for observed and missing Y. Thereby $\hat{\beta}$ is used to predict the known values $[P_{\Omega}(Y)]_i$ and $\beta^{\star}$ is used to predict the missing ones $[P_{\Omega^c}(Y)]_i$.
iv) For each prediction for a missing value $[P_{\Omega^c}(Y)]_i$ find a small set (typically 3, 5 or 10, see van Buuren, 2018) of the closest predicted values (based on $\hat{\beta}$) for the observed values.
v) Randomly decide on one of the elements in the set and impute it with its corresponding observed value.

This ensures that only values observed in the domain of the variable are imputed. As summarized in van Buuren & Groothuis-Oudshoorn (2011): "Its main virtues are that imputations are restricted to the observed values and that it can preserve non-linear relations even if the structural part of the imputation model is wrong. It is a good overall imputation method."

For binary categorical data, a popular imputation method is based on logistic regression, which was pointed out by Rubin (1987), consisting of the following steps:

i) First, estimate for the observed values a logistic regression model for $P_{\Omega}(Y)$ and obtain estimated $\hat{\beta}$ and the corresponding covariance matrix V by iteratively reweighted least squares (van Buuren, 2018).
ii) Draw a new parameter set $\beta^{\star} \sim N(\hat{\beta}, V)$.
iii) For each missing value $[P_{\Omega^c}(Y)]_i$ predict $logit^{-1}(X_i\beta^{\star})$ and randomly draw a number $u_i \sim U(0, 1)$.
iv) If $u_i > logit^{-1}(X_i\beta^{*})$ then impute $[P_{\Omega^c}(Y)]_i = 0$, otherwise $[P_{\Omega^c}(Y)]_i = 1$.

These steps are performed repeatedly n_{imp} times to obtain the n_{imp} data sets of multiple imputation.

After multiple imputation, these different n_{imp} data sets can be clustered by applying the k-prototypes algorithm. Finally these n_{imp} not necessarily equal partitions have to be aggregated to determine one partition for the data set with missing values.

3.2 Pooling of Partitions

The aim is the aggregation of the n_{imp} partitions to obtain exactly one partition for the data set with missing values. Since every imputed mixed-type data set is clustered by the k-prototypes algorithm, every object in the data set has n_{imp} (possibly different) cluster assignments $S_i = (s_{i1}, \dots, s_{in_{\text{imp}}})$, where $s_{il} \in \{1, \dots, k\}$ and $i \in \{1, \dots, n\}$. Additionally, it is conceivable that the clusters in the different imputed data sets can be labeled differently. Pooling the cluster results of the multiple imputed data sets finds a clustering that matches as much as possible with the n_{imp} given partitions.

Similar to bagged clustering (Leisch, 1999), Gionis et al. (2005) have shown several approaches of which the agglomerative algorithm is adapted in the following. As a standard bottom-up algorithm, every object S_i is placed into a singleton cluster G_i at first. By adapting the hierarchical clustering method with average linkage (Contreras & Murtagh, 2015), the pair of clusters (G_i, G_j) with the smallest average distance d_{AL} between the associated objects is linked together, where

$$d_{\text{AL}}(G_i, G_j) = \frac{1}{||G_i|| \times ||G_j||} \sum_{S_a \in G_i, S_b \in G_j} d_{\text{pool}}(S_a, S_b).$$

Therefore, the distance between two objects in the presented use case of this paper is defined as the proportion of different cluster assignments

$$d_{\text{pool}}(S_a, S_b) = \frac{1}{n_{\text{imp}}} \sum_{l=1}^{n_{\text{imp}}} \delta(S_{al}, S_{bl}), \text{ where } \delta = \begin{cases} 0, & S_{al} = S_{bl}, \\ 1, & S_{al} \neq S_{bl}. \end{cases}$$

In summary, it can be stated that (singleton) clusters are merged together if the average distance between objects of the n_{imp} partitions are sufficiently similar. That is, clusters are merged if the average distance of the closest pair of clusters is less than 0.5. Otherwise (e.g., there is no pair of clusters with average distance smaller than 0.5) the linkage of clusters cannot improve the partition and the algorithm stops.

Since estimating the number of clusters k is beyond the scope of our simulation study, the cluster partitioning for the data set with the missing values is obtained by cutting the resulting tree into the desired group size (i.e., the desired number of clusters in the presented use case). For additional information on the selection of a suitable number of clusters for mixed-type data see Aschenbruck & Szepannek (2020).

4 Simulation Study

To evaluate the previously presented approaches for clustering mixed-type data with missing values and imputation, a simulation study is conducted. Similar to Jimeno et al. (2021), several data settings, hereinafter referred to as scenarios, were investigated. In order to compare the various approaches to cluster mixed-type data with missing values, the considered simulation settings are outlined in the following.

Table 1 Features and associated feature specifications used to generate the data for the simulation study as well as the color coding (light gray, gray and brown) of the individual feature settings for the figures

Feature	Feature specification			Short notation
	Color coding			
	■	■	■	
Number of clusters	2	4	8	`nC`
Clusters of equal size	`TRUE`	`FALSE`		`symm`
Number of variables	4	8		`nV`
Ratio of factor to numerical variables	0.25	0.5	0.75	`fac_prop`
Overlap between cluster groups	0	0.05	0.1	`overlap`

4.1 Data Simulation

Simulating different data scenarios, the five features *number of clusters*, *clusters of equal size*, *number of variables*, *ratio of factor variables to numerical variables*, and *overlap between cluster groups* are varied in parameter values. In Table 1, these features and their corresponding parameter values for the simulation are listed. Additionally, the color coding for Fig. 1 and the short identification name used among others in Figs. 1 and 3 is presented in the table. Since a full factorial design is used, there are $3 \times 2 \times 2 \times 3 \times 3 = 108$ different scenarios in the conducted simulation study. The selection of the considered features follows the characteristics of the simulation study of Dangl & Leisch (2020) and is extended with respect to the ratio of the variable types (Jimeno et al., 2021).

The clusters are defined by the structure of the feature settings. Each variable can be either active or inactive. For the numerical variables, *active* means drawing values from the normal distribution $X_1 \sim N(\mu_1, 1)$, where μ_1 is a randomly determined value, and *inactive* means drawing from $X_0 \sim N(\mu_0, 1)$ with $\mu_0 = 2q_{1-\frac{v}{2}} - \mu_1$, where q_α is the α-quantile of $N(\mu_1,1)$ and $v \in \{0.05, 0.1\}$. This results in an overlap of v for each of the two normal distributions. If there is an overlap of $v = 0$, the inactive variable is drawn from $N(\mu_1 - 10,1)$. On the other hand, every factor variable has two levels l_0 and l_1. The probability for drawing l_0 for an active variable in a cluster is $v \in \{0, 0.05, 0.1\}$ and $1 - v$ for level l_1. For an inactive variable, the probability for l_0 is $(1 - v)$ and v for l_1.

4.2 Generating Different Types of Missing Values

According to Little & Rubin (2019), there are generally three different mechanisms that underly missing data and whose impact is analyzed in our simulation study. First, there are missing values that are neither conditioned on the value of the missing variable, nor on the other observed variables in the data set. These missing values are called *missing completely at random* (MCAR). In contrast, the so-called *missing at random* (MAR) mechanism means, that the missings depend only on the observed variables. If the mechanism causing missing data is neither MCAR nor MAR, it is called MNAR (*missing not at random*), i.e., the occurrence of a missing value depends also on the underlying value itself (Carpenter & Kenward, 2012).

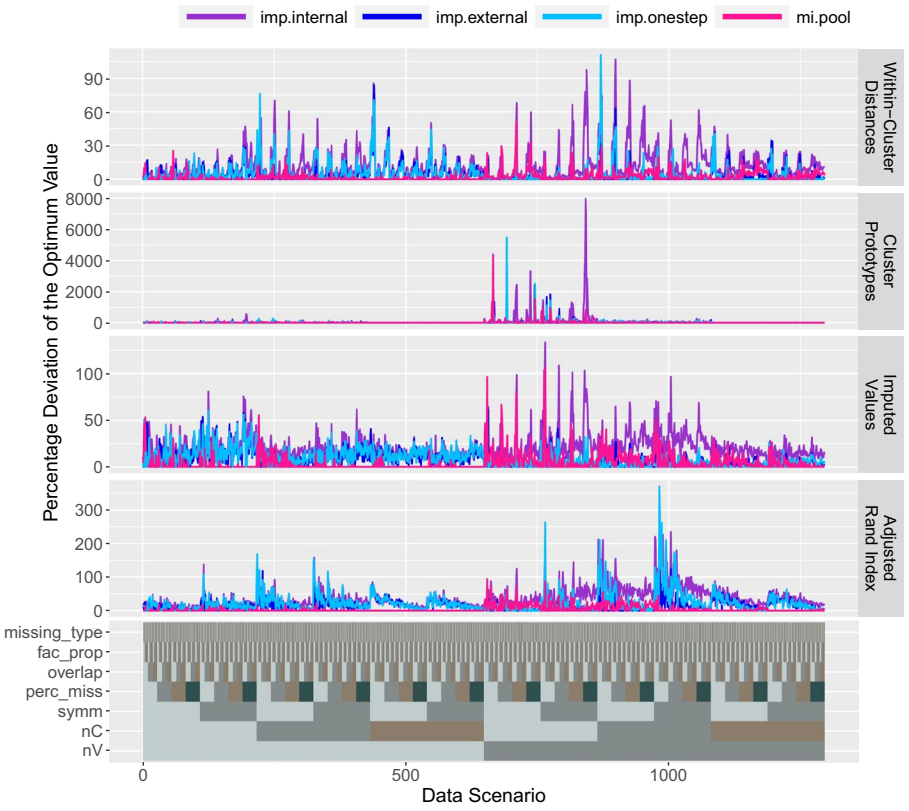

Fig. 1 The upper main plot shows the percentage deviation of the optimum value for the mean value of each scenario over the $N = 50$ iterations for all four evaluation criteria. The settings of the parameters of each scenario are displayed in the lower plot with the color coding presented in Table 1 (Section 4.1) and Table 2 (Section 4.2)

Table 2 Features and associated feature specifications used to generate the missing values for the simulation study as well as the color coding (light gray, gray, brown and dark gray) of the individual feature settings for the figures

Feature	**Feature specification**				**Short notation**
	Color coding				
Type of missing values	MCAR	MAR	MNAR		`missing_type`
Incomplete data (%)	0.05	0.1	0.2	0.4	`perc_miss`

The features settings used in the simulation study to generate missing values are given in Table 2. In addition to the feature *type of missing values*, the parameter `perc_miss` gives the percentage of observations with at least one missing value. In summary, when generating the missing values, each observation is first assigned to a possible

pattern of missing values, that indicates which variable(s) should contain (a) missing value(s). For each scenario and based on the number of variables, up to $\texttt{nV}^2/3$ different patterns are generated randomly. Afterwards, the specified proportion of incomplete data `perc_miss` is generated for each data block per pattern. For the missing mechanism MCAR, the observations which will not remain complete, are determined randomly. On the other hand, for the missing mechanisms MAR and MNAR the probability for amputing values depends on the observed values: For MAR, the variables which remain complete determine the missingness probability, and for the MNAR mechanism, the values of the variable to be amputed affects the chance to be deleted (van Buuren, 2018). With these two additional study parameters the full factorial experimental design increases to $108 \times 3 \times 4 = 1296$ scenarios.

4.3 Evaluation Aspects

Before the evaluation aspects are explained in more detail, a note on the procedure in general is given: Since the simulation study is performed on artificially generated data sets with n observations, where also the missing values are simulated, the real values of the missing values are known. In the following, the complete data (before the generation of the missing values) will be referred to as *original data*. Following this notation, first the original data is clustered using the k-prototypes algorithm. In this way, the so-called *original prototypes* and *original cluster assignments* are obtained and defined as reference values.

To evaluate the efficiency and the goodness of the different strategies for imputing and clustering mixed-type data with missing values, four different criteria on different aspects are considered.

i) *within-cluster distances of the partition*
   This evaluation aspect analyzes the aim of the k-prototypes algorithm to minimize the within-cluster distances. To accomplish the latter, the sum of the distances between the n observations of the original data set and the associated prototype based on the cluster analysis of the incomplete data is examined and thus the goodness of the partition can be evaluated.
ii) *determined prototypes*
   This aspect evaluates the quality of the determined prototypes. For this purpose, the distance between the prototypes based on clustering the incomplete data and the original prototypes is determined. Since these clusters are not necessarily labeled identically, the minimum distance of all permutations of the distances is considered in each case.
iii) *imputed values for missings*
   To determine the quality of the imputed values, the distance between the original data and the data with imputed values is evaluated.
iv) *cluster assignments of the clustered objects*
   Finally, the adjusted Rand Index (Hubert & Arabie, 1985) is determined to the original cluster assignments of the original data and the resulting partition based on the analysis of the incomplete data with missing values.

The distance determination for all evaluation aspects is realized with the distance measure shown in (1) in Section 2.1.

4.4 Execution of the Simulation Study

The simulation study is conducted using the open source software R (R Core Team, 2020). The underlying R code is available at GitHub[1]. For each of the 1296 predefined scenarios, an original data set is generated and clustered with the k-prototypes algorithm using the `validation_kproto` function (Aschenbruck & Szepannek, 2020) of the R package *clustMixType* (Szepannek, 2018; Szepannek & Aschenbruck, 2021). The resulting so-called original cluster prototypes and cluster assignments are used in the following for benchmarking as described in Section 4.3. Afterwards the missing values are generated as described in Section 4.2 with the function `ampute` of the R package *mice* (van Buuren & Groothuis-Oudshoorn, 2021).

For conducting cluster analysis and imputation on the incomplete data, the following four presented methods are compared. The first three named variants are based on the approach derived in Section 2 and differences between those are presented in more detail in Section 2.3. The last approach is based on Section 3.

i) `kproto(..., na.rm = "imp.internal")`
   The function `kproto` from the R package *clustMixType* is executed, and within the algorithm the missing values are imputed with the current prototypes after each assignment of new prototypes.
ii) `kproto_imp.external(...)`
   In this variation, the function `kproto` is executed as well, but the missing values are ignored during the execution of the cluster algorithm. After a partition is found, the (former) missing values are imputed with the values of the corresponding prototypes. Afterwards, the `kproto` function is executed again on the imputed data set, as long as the imputed values are changing (the maximum number of iterations with an imputation after each execution is 100).
iii) `kproto(..., na.rm = "imp.onestep")`
   Again, the function `kproto` is applied ignoring the missing values during the execution of the cluster algorithm, but here it is applied only once (meaning the just presented *strategy ii)* is stopped after the first step). So finally, after determining the partition, the missing values are imputed with the values of the corresponding cluster prototypes.
iv) `kproto_mi.pool(...)`
   In this approach, multiple imputation is used to create $n_{\text{imp}} = 5$ data sets by executing R function `mice` of R package *mice* (van Buuren & Groothuis-Oudshoorn, 2021). Hereof, internally the functions `mice.impute.pmm` (set size of 5 closest as recommended by van Buuren 2018) and `mice.impute.logreg` (van Buuren & Groothuis-Oudshoorn, 2011; White et al., 2010) are used for imputation depending on the variable type. The resulting n_{imp} data sets are clustered by using the `kproto` function. Afterwards, the results are pooled as described in Section 3.2 using the R function `hclust`. Finally, the missing values are imputed with the values of the prototypes of the pooled partition.

In order to reduce the influence of the randomness by generating the artificial data, $N = 50$ repetitions are performed for each scenario. Additionally, it is well known that both algorithms, k-prototypes and k-means, are based on the random selection of initial cluster centers or prototypes, respectively. For the k-means algorithm, the optimization of the

[1] https://github.com/rabea-a/JClassif_ImputationClusterMixedType.

Pseudo-Code 1 Simulation study.

```
for n in 1:1296 do
    params <- trials_design[n,]   // set specifications of full
                                     factorial design
    for i in 1:50 do
        set.seed(i)                          // for reproducibility
        origin <- create_mtd(params)    // simulate artificially
                                           mixed type data

        // determine optimal partition for original data:
        kpres_sil <- validation_kproto(method = "silhouette",
                         data = origin k = 2:10, nstart = 3)
        // simulate random pattern for missings and ampute
           values
        my_pattern <- expand.grid(rep(list(c(0,1)),
                        nV))[sample(2:(nV^2-1),
                        size = sample(1:floor(nV^2/3))),]
        data_NA <- mice::ampute(data = origin, params)

        // imputation and clustering:
        // apply the 4 methods of interest to the identical
           data set and use the same values for the input
           parameters of the cluster algorithm
        // use for all 4 methods: kpres_sil$k_opt,
           kpres_sil$lambda
        kpres_internal <- kproto(data_NA,
                                 na.rm = "imp.internal")
        kpres_external <- kproto_imp.external(data_NA)
        kpres_one <- kproto(data_NA, na.rm = "imp.onestep")
        kpres_mipool <- kproto_mi.pool(data_NA, n_imp = 5)

        for every imputation and clustering method do
            eval_wcd <- eval_within(origin = origin,
                          kpres = kpres_method)
            eval_c <- eval_protos(protos = kpres_method$centers,
                        protos_origin = kpres_sil$centers,
                        lambda = kpres_sil$lambda,
                        k_opt = kpres_sil$k_opt)
            eval_iv <- eval_iv(origin = origin,
                         x_iv = kpres_method$data,
                         lambda = kpres_sil$lambda)
            eval_r <- fossil::adj.rand.index
                        (kpres_method$cluster,
                        kpres_sil$centers)
```

initialization has already been in the focus of some research publications (see, e.g., Peña et al., 1999; Fränti & Sieranoja, 2019). The transfer of these (or similar) strategies to the k-prototypes algorithm for mixed-type data is not the scope of this article, but nevertheless, it is an interesting field for future research. Therefore, the influence of the randomly generated initial prototypes in the conducted simulation study is minimized using the built-in repetition of the R function `kproto` by setting the parameter `nstart=3`.

The evaluation criteria presented in Section 4.3 are implemented as functions named `eval_within_dist(origin, kpres)` (evaluation of the *within-cluster distances of the partition*), `eval_protos(protos, protos_origin, lambda, k_opt)` (*determined prototypes*) and `eval_iv(origin, x_iv, lambda)` (*imputed values for missings*). For evaluation of *cluster assignments of the clustered objects*, the function

`adj.rand.index` of R package *fossil* (Vavrek, 2011) is used. The implementation of the simulation study is shown in Pseudo-Code 1 to summarize the workflow.

4.5 Results of the Simulation Study

An overview of the values obtained in the simulation study is given in Fig. 1. The mean value was calculated for each evaluation aspect over the $N = 50$ iterations and for each examined imputation/clustering method and for each scenario. In the upper main plot the percentage deviation from the best observed mean value for each data set is shown for the four evaluation aspects *within-cluster distances*, *cluster centers*, *imputed values* and *adjusted Rand index*. Below, the parameters of the data set are illustrated with the colors light gray, gray, brown and dark gray: `missing_type` (MNAR, MAR, MCAR), `perc_miss` (0.05, 0.1, 0.2, 0.4), `overlap` (0, 0.05, 0.1), `fac_prop` (0.25, 0.5, 0.75), `symm` (true, false), `nC` (2, 4, 8) and `nV` (4, 8) (cf. Sections 4.1 and 4.2 for details and in particular Tables 1 and 2 for color coding). Particularly strong fluctuations can be seen in the evaluation of the cluster prototypes. Even apart from some clear outliers, some mean values are more than 1000% larger than the best observed evaluation value. It is obvious that the methods handle the different scenarios differently well. The results indicate that the multiple imputation approach (*mi.pool*, pink) is worse at dealing with a larger number of variables (`nV=8`, level 2, dark gray). It also turns out that the *imp.onestep* (light blue) performs worse with a smaller number of missing values (`perc_miss=0.05`, level 1, light gray) than with more missing values. Therefore, it makes sense to also investigate the influence of the different data set parameters on the imputation and clustering quality. Because of the huge amount of data, it is hard to identify every difference between the four approaches in Fig. 1. In the following, the four methods are directly compared by ranks, first without considering the data set parameters (Fig. 2). The reason for considering the ranks is to avoid a biased comparison due to the variability between the different scenarios and iterations.

Therefore, for each evaluation aspect (within-cluster distances, prototypes, imputed values and cluster assignment) ranks between the four methods were computed and the percentages over all iterations and data sets were determined. The best method (as defined for the different aspects in Section 4.3) for handling missing values in a clustering task is rated with rank 1. The average rank is assigned if the values are equal. The figure is based on all results for the 50 iterations for all of the 1296 data/missing scenarios. Due to a better readability, only areas larger than 10% were labeled. It is easy to identify, that method *imp.internal* is the worst compared to the other three methods. It also can be stated that the results based on *mi.pool* are most often ranked best for all aspects studied apart from computation time. The methods *imp.external* and *imp.onestep* have similarly rated results regarding the four evaluation criteria. Remarkably, both methods have less often the worst results among the evaluated criteria *within-cluster distances*, *cluster prototypes* and *imputed values*, whereas the approach based on multiple imputation is more often the worst of all methods. Only in the evaluation of cluster assignments with the adjusted Rand index, the *mi.pool* method seems to have a slight advantage. In conclusion, that means, *mi.pool* is more unstable in the ranking of the observed results, which seems to make sense, as additional randomness is introduced by this imputation strategy. Considering the evaluation over the computation time, it can be stated that *imp.onestep* is by far the best. Given that very similarly rated results are obtained as with *imp.external*, *imp.onestep* should be preferred.

If computation time is not a limiting factor, it would be interesting to know which method should be preferred in which situation. For analyzing the effects of individual data

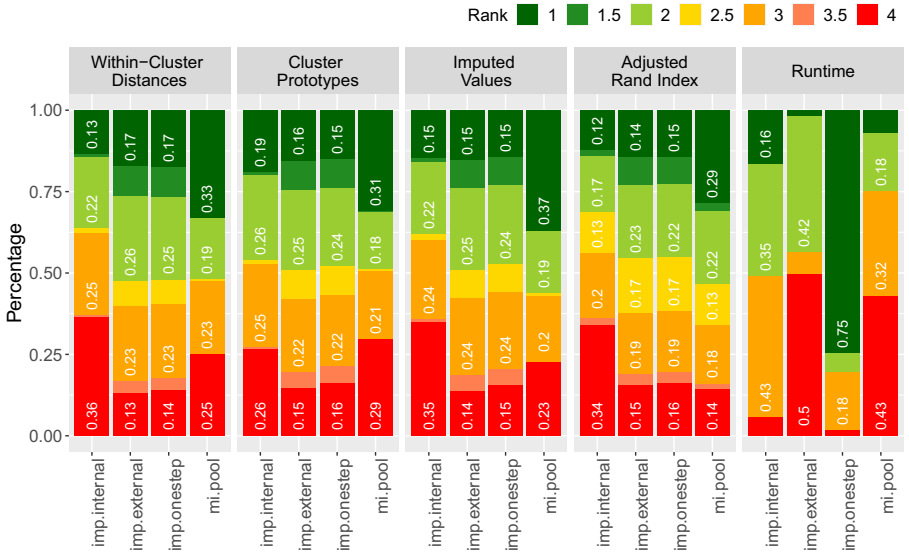

Fig. 2 Presentation of the percentage of ranks for the four different aspects of evaluation and the observed runtime based on every data/missing scenario and all $N = 50$ iterations of the simulation study

scenarios on the goodness of the applied methods, logistic regression models were fitted (Agresti, 2007). The aim is to get an idea of the impact of the different data scenario parameters on the conductivity of the different methods. The target variable $B_i \in \{0, 1\}$ denotes whether a method is the (not necessarily sole) best (1) or not (0), such that

$$P(B_i = 1) = \frac{\exp(a_i^T \beta)}{1+\exp(a_i^T \beta)} \tag{7}$$

$$\Leftrightarrow \text{Odds}(B_{1/0}) = \frac{P(B_i=1)}{P(B_i=0)} = \exp(a_i^T \beta) = \prod_p \exp(a_{ip}^T \beta_p), \tag{8}$$

where a_i displays the data scenario parameters as binary dummy variables. The following reference categories are used for the analyzed variables: `nV=4`, `nC=2`, `symm=FALSE`, `perc_miss=0.05`, `overlap=0`, `fac_prop=0.5`, `missing_type=MCAR`. Figure 3 shows the exponentiated coefficients $\exp(\beta)$ of the logistic regressions for each evaluation aspect in a separate plot and each method is displayed with an unique color. A value greater than 1 means an increase in the odds, which corresponds to an increasing probability that the examined method on a data set with this feature setting is better with respect to the reference category of this feature. A value of less than 1 indicates the opposite, namely that the method for this parameter value is worse than for the reference value. Since coefficients with values equal to 1 or with the value of 1 in the associated confidence interval represent no change to the reference category and are therefore negligible, these coefficients are shown transparent in Fig. 3.

Roughly speaking, similar results for the coefficients can be seen for the four different evaluation criteria. Surprisingly, the different missing value mechanisms do not play a crucial role for the choice of a preferable strategy, no matter for which evaluation aspect. The proportion of categorical variables is also rather negligible in

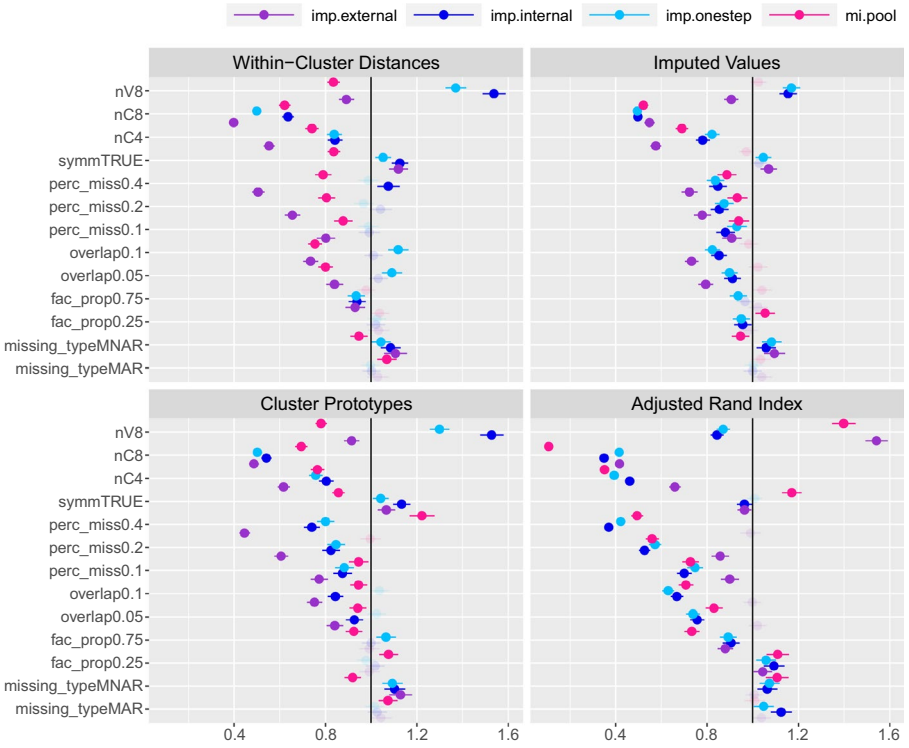

Fig. 3 Presentation of the exponentiated logistic regression coefficients for parameters of the data set features for each evaluation aspect and each imputation/clustering method, where coefficients whose confidence intervals contain 1 are displayed in transparent

comparison to other data set parameters. For the other influencing variables namely *number of variables*, *number of clusters*, *equal cluster sizes*, *proportion of missing values* as well as *overlap of clusters*, however, an effect seems to be noticeable. When having a closer look at the evaluation of the within-cluster distances, it is remarkable that for a bigger overlap and more missing values the methods *imp.internal* (purple) and *mi.pool* (pink) perform worse than the reference category, whereas the results of the methods *imp.external* (darkblue) and *imp.onestep* (light blue) are not influenced by these variables or even slightly better than the reference category. Also the increasing number of missing values and overlapping decreases the rating for evaluation the imputed values and the cluster prototypes for the methods *imp.onestep*, *imp.internal* and *imp.external*. The strongest differences can be seen in the variables *number of variables* and *symmetry*. For the evaluation aspects *within-cluster distance*, *imputed values* and *cluster prototypes*, it can be seen that, e.g., a higher number of variables worsens the results for the methods *imp.external* and *mi.pool* and improves the results for the methods *imp.internal* and *imp.onestep*. The opposite is true for the evaluation by *adjusted Rand index*. The increase of the number of clusters leads to a worse result for each evaluation criteria for each examined method.

Table 3 Descriptive overview on the Titanic data set

Feature	Data type	Missing values	Mean	Levels		
country	categorical	564	—	UK	USA	other
				(236)	(299)	(210)
sex	categorical	0	—	female	male	
				(466)	(843)	
age	numerical	263	29.881	—		
embarked	categorical	2	—	S	C	Q
				(914)	(270)	(123)
fare (in £)	numerical	1	33.295	—		
passenger class	categorical	0	—	lower	middle	upper
				(709)	(277)	(323)
survived	categorical	0	—	no	yes	
				(809)	(500)	

5 Application to a Real-world Problem

In order to provide an application of the adapted method, it is applied to the well-known mixed-type Titanic data set (Eaton & Haas, 1994), where the data is accessed from the OpenML platform (Vanschoren et al., 2013) by using the R package *OpenML* (Casalicchio et al., 2017). The data to be clustered contain information on 1309 titanic passengers in seven features, which are displayed in Table 3 together with the mean value (for numeric features) or the levels with frequencies (for the categorical ones). The presented feature *country* summarizes the variable *home.dest*, which contains information on Cities of the home destination of the passengers. The aim of the cluster analysis is to identify interesting structures, considering not only the 684 complete cases out of the total number of 1309 available observations. Derived from the findings of the simulation study, we apply the *one step imputation*, whereby the number of clusters to be determined is validated with the Silhouette Index v_{sil} (this choice is based on the results in Aschenbruck & Szepannek, 2020).

The application of the internal validation index determines an index-optimal number of clusters of two ($v_{\text{sil}} = 0.627$). The partition resulting from the application of the k-prototypes algorithm using *one step imputation* is shown in Fig. 4 and the respective prototypes in Table 4. The characteristics of the clusters are visualized by displaying one line per passenger, whereby theses lines are jittered horizontally for a better visualization of the categorical features. In Cluster 1, it can be seen that a large proportion of passengers are men from the UK. Although most passengers in Cluster 1 boarded the Titanic in Southampton, there are also some boardings in Cherbourg and Queenstown. Mostly rather low prices were paid for tickets, which is reflected in the occupancy of the lowest passenger class predominantly. Although a certain proportion survives the sinking of the Titanic, three quarters of the passengers from Cluster 1 die. In contrast, Cluster 2 represents almost exclusively passengers from the USA, and two-thirds of them are female. Comparatively higher prices were paid by passengers grouped in Cluster 2 and consequently an almost exclusive accommodation in the upper passenger class can be observed. The majority of passengers who are grouped in this cluster survived the sinking of the Titanic.

Since the application to the real-world data set is done to present the usage of the derived imputation strategy, the focus in the following is not on further interpretation of

Fig. 4 Partition of two clusters resulting from the application of the k-prototypes algorithm with the usage of the one step imputation

the identified structures, but on the imputation results. However, it must be noted that neither missing values are known and therefore the evaluation of imputations is non-trivial, nor some kind of a true partition is given for this kind of data. Therefore, we will consider the obtained imputations by their plausibility. Since almost exclusively, the missing values occur in the categorical variable *country* and the numerical variable *age*, a closer look at these variables follows. As Table 3 shows, 564 missing values occur in the categorical feature *country*, which means that for only 57% of the passengers the home country is known. For 534 passengers of the Titanic, the home country was imputed as the UK and for 30 persons as the USA. This is remarkable because the most often known home country is the USA and naive imputation strategies would therefore have used this specification in (at least) the majority of imputations. However, this is not plausible due to the location of the Titanic's first stop on the maiden voyage in Southampton, UK. Furthermore, it should be kept in mind that ship voyages at that time were often used for the purpose of emigration (Jones, 1976), and in this context it is plausible that a large part of the passenger's home country would not be imputed with the USA. The age is not known for 263 cases. Through the *one step imputation*, an age of 28.164 years was determined for 250 passengers and 36.558 years for the other 13 passengers. The naive imputation of the mean value of the variable age for all missing ages would have resulted in a value of about 30 years. Especially for the passengers with higher priced tickets, the imputation of a higher age seems plausible. These passengers might not have embarked on the Titanic for reasons of emigration, but to experience the luxury and the idealistic value of the voyage. Therefore, these passengers are presumably already wealthier and thus more likely to be older.

In summary, it can be seen that an application to real-world data is possible without any problems. The resulting values for imputation and also the cluster structure seem plausible, although one cannot determine the correctness as usual in case of missing data. Additionally, it should be emphasized that using the proposed method enables to process the data

Table 4 Prototypes of two clusters resulting from the application of the k-prototypes algorithm with the usage of the one step imputation

	Country	Sex	Age	Embarked	Fare (in £)	Passenger class	Survived
Cluster 1	UK	male	28.164	S	17.043	lower	no
Cluster 2	USA	female	36.558	C	110.690	upper	yes

from all 1309 passengers and not only from the 684 complete cases. Beyond this gain of information for cluster determination, a cluster assignment for all given observations can be derived, i.e., for twice as many cases as without applying an appropriate imputation method.

6 Discussion and Conclusion

The aim of this study was to investigate the handling of missing values in the cluster analysis of mixed-type data. The focus was on the k-prototypes algorithm, an extension of the widely known k-means algorithm. For handling missing values while clustering numerical and categorical data, it has been demonstrated, that it is possible to extend the k-POD imputation strategy to mixed-type data and the k-prototypes algorithm. The following variants of this approach were examined: imputation over the values of the corresponding prototypes in each iteration step of the k-prototypes algorithm (*internal imputation*), repeated imputation only after execution of the k-prototypes algorithm (*external imputation*) and the associated fast variant with only one execution of the k-prototypes algorithm and subsequent imputation (*one step imputation*). Additionally, an approach based on multiple imputation and cluster aggregation with an agglomerative algorithm was presented (*multiple imputation and pooling*).

For the *internal imputation* it has been shown in the simulation study, that the imputation of the missing values within the k-prototypes algorithm is not preferable compared to the other examined strategies. Further, the approach of *repeated external imputation* showed almost the equally rated results as the remarkably faster *one step imputation*. This is observed for all four evaluation criteria concerning partition, prototypes, imputed values and cluster assignment, so there is no major benefit to use the time consuming *external imputation*. The approach based on *multiple imputation and subsequent cluster aggregation* more often achieves the best result, but in comparison to the *external* and *one step imputation* also more often the worst ones. Surprisingly, even in the case of evaluation of the imputed values, the method with *multiple imputation and pooling* yields more frequently poorly evaluated results than the two strategies based on the imputation with prototype values obtained by the k-prototypes algorithm. Furthermore, this *multiple imputation approach* is, similar to the *external imputation*, not nearly as time-efficient as the *one step imputation*. Especially in the case of a larger number of variables and a higher proportion of missing values, the *one step imputation* is to be preferred. Furthermore, it could be determined, that the underlying missing mechanisms only have very little impact in the choice of a preferable strategy. In summary, the *one step imputation* developed throughout this paper is a straightforward and time-saving imputation method that can be applied reliably and with satisfactory results in a variety of scenarios. At last, it was shown that this method leads to meaningful results even in an application to real-world data, and that the imputation strategy is not merely a theoretically useful procedure which is reflected by plausible imputed values.

It must be noted that the elaborated findings are restricted to the examined cluster structure. The approach of multiple imputation with subsequent cluster aggregation can be applied to

any cluster structures and algorithms and is therefore much more versatile. In future work, it would be interesting to investigate how cluster stability differs among the different approaches (Hennig, 2007). An extension of the investigation to further real-world data is also desirable. However, it is important to focus on the problem of the non-trivial evaluation, which generally dominates the analysis of data with missing values.

Appendix: Proofs of Theorems

A.1 Proof of Theorem 1 for Numerical Values

In the following, the abbreviated notation $K_i(C^{(t)}, W^{(t)}) = K_i^{(t)}$ is used for clarity. For all numerical values $i \in \{1, \dots, n\}, j \in \{1, \dots, l\}$ is:

a) $[\tilde{X}_i]_j = [P_\Omega(X_i)]_j + [P_{\Omega^C}(K_i(C^{(t)}, W^{(t)}))]_j = [P_\Omega(X_i)]_j + [P_{\Omega^C}(K_i^{(t)})]_j$,
b) $[K_i(C, W)]_j = [P_\Omega(K_i(C, W))]_j + [P_{\Omega^C}(K_i(C, W))]_j = [P_\Omega(K_i)]_j + [P_{\Omega^C}(K_i)]_j$,
c) $[P_\Omega(B_i)]_j [P_{\Omega^C}(C_i)]_j = 0$ with $B_i, C_i \in \mathbb{M}^{1\times m}$.

Showing the equality of the formula above for numerical values:

$$
\begin{aligned}
\sum_{i=1}^{n}\sum_{j=1}^{l}\left([\tilde{X}_i]_j - [K_i(C,W)]_j\right)^2 &\overset{a),b)}{=} \sum_{i=1}^{n}\sum_{j=1}^{l}\left([P_\Omega(X_i)]_j + [P_{\Omega^C}(K_i^{(t)})]_j - [P_\Omega(K_i)]_j - [P_{\Omega^C}(K_i)]_j\right)^2 \\
&= \sum_{i=1}^{n}\sum_{j=1}^{l}[P_\Omega(X_i)]_j^2 + [P_{\Omega^C}(K_i^{(t)})]_j^2 + [P_\Omega(K_i)]_j^2 + [P_{\Omega^C}(K_i)]_j^2 \\
&\quad + 2[P_\Omega(X_i)]_j[P_{\Omega^C}(K_i^{(t)})]_j - 2[P_\Omega(X_i)]_j[P_\Omega(K_i)]_j \\
&\quad - 2[P_\Omega(X_i)]_j[P_{\Omega^C}(K_i)]_j - 2[P_{\Omega^C}(K_i^{(t)})]_j[P_\Omega(K_i)]_j \\
&\quad - 2[P_{\Omega^C}(K_i^{(t)})]_j[P_{\Omega^C}(K_i)]_j + 2[P_\Omega(K_i)]_j[P_{\Omega^C}(K_i)]_j \\
&\overset{c)}{=} \sum_{i=1}^{n}\sum_{j=1}^{l}[P_\Omega(X_i)]_j^2 + [P_{\Omega^C}(K_i^{(t)})]_j^2 + [P_\Omega(K_i)]_j^2 + [P_{\Omega^C}(K_i)]_j^2 \\
&\quad - 2[P_\Omega(X_i)]_j[P_\Omega(K_i)]_j - 2[P_{\Omega^C}(K_i^{(t)})]_j[P_{\Omega^C}(K_i)]_j \\
&= \sum_{i=1}^{n}\sum_{j=1}^{l}[P_\Omega(X_i)]_j^2 - 2[P_\Omega(X_i)]_j[P_\Omega(K_i)]_j + [P_\Omega(K_i)]_j^2 \\
&\quad + [P_{\Omega^C}(K_i)]_j^2 - 2[P_{\Omega^C}(K_i^{(t)})]_j[P_{\Omega^C}(K_i)]_j + [P_{\Omega^C}(K_i^{(t)})]_j^2 \\
&= \sum_{i=1}^{n}\sum_{j=1}^{l}\left([P_\Omega(X_i)]_j - [P_\Omega(K_i)]_j\right)^2 + \left([P_{\Omega^C}(K_i)]_j - [P_{\Omega^C}(K_i^{(t)})]_j\right)^2 \\
&= \sum_{i=1}^{n}\sum_{j=1}^{l}\left([P_\Omega(X_i)]_j - [P_\Omega(K_i(C,W))]_j\right)^2 \\
&\quad + \left([P_{\Omega^C}(K_i(C,W))]_j - [P_{\Omega^C}(K_i(C^{(t)}, W^{(t)}))]_j\right)^2.
\end{aligned}
$$

A.2 Proof of Theorem 1 for Categorical Values

In the following, the abbreviated notation $K_i(C^{(t)}, W^{(t)}) = K_i^{(t)}$ is used for clarity. For all categorical values $i \in \{1, \dots, n\}, j \in \{l+1, \dots, m\}$ is:

d) $\tilde{X} := \left[[P_{\Omega}(X_i)]_j \; \forall (i,j) \in \Omega \wedge [P_{\Omega^C}(K_i(C^{(t)}, W^{(t)}))]_j \; \forall (i,j) \notin \Omega\right]$,
e) $K(C,W) := \left[[P_{\Omega}(K_i(C,W))]_j \; \forall (i,j) \in \Omega \wedge [P_{\Omega^C}(K_i(C,W)]_j \; \forall (i,j) \notin \Omega\right]$,
f) $\delta(\text{„NA“,„NA“}) = 0$.

Showing the equality of the formula above for categorical values:

$$
\begin{aligned}
& \lambda \sum_{i=1}^{n} \sum_{j=l+1}^{m} \delta\left([P_{\Omega}(X_i)]_j, [P_{\Omega}(K_i(C,W))]_j\right) + \delta\left([P_{\Omega^C}(K_i(C,W))]_j, [P_{\Omega^C}(K_i(C^{(t)}, W^{(t)}))]_j\right) \\
= \; & \lambda \sum_{i=1}^{n} \sum_{j=l+1}^{m} \delta\left([P_{\Omega}(X_i)]_j, [P_{\Omega}(K_i(C,W))]_j\right) + \delta\left([P_{\Omega^C}(K_i(C^{(t)}, W^{(t)}))]_j, [P_{\Omega^C}(K_i(C,W))]_j\right) \\
\overset{d),e),f)}{=} \; & \lambda \sum_{i=1}^{n} \sum_{j=l+1}^{m} \delta\left([\tilde{X}_i]_j, [K_i(C,W)]_j\right).
\end{aligned}
$$

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability The datasets generated during and/or analyzed during the simulation study are available from the corresponding author on reasonable request.

Declarations

Conflict of Interest The authors declare no competing interests.

Ethical Conduct This research does not contain any studies with human participations or animals performed by any of the authors.

References

Agresti, A. (2007). *An introduction to categorical data analysis*, 2nd edn. New York: Wiley. https://doi.org/10.1002/0470114754.

Ahmad, A., & Khan, S. (2019). Survey of state-of-the-art mixed data clustering algorithms. *IEEE Access*, 31883–31902. https://doi.org/10.1109/ACCESS.2019.2903568.

Aschenbruck, R., & Szepannek, G. (2020). Cluster validation for mixed-type data. *Archives of Data Science Series A*, *6*(1), 1–12. https://doi.org/10.5445/KSP/1000098011/02.

Audigier, V., & Niang, N. (2020). Clustering with missing data: which equivalent for rubin's rules? https://doi.org/10.48550/arXiv.2011.13694.

Basagaña, X., Barrera-Gómez, J., Benet, M., Antó, J. M., & Garcia-Aymerich, J. (2013). A framework for multiple imputation in cluster analysis. *American Journal of Epidemiology*, *177*(7), 718–725. https://doi.org/10.1093/aje/kws289.

Carpenter, J. R., & Kenward, M. G. (2012). *Multiple imputation and its application.* John Wiley & Sons Ltd. https://doi.org/10.1002/9781119942283.

Casalicchio, G., Bossek, J., Lang, M., Kirchhoff, D., Kerschke, P., Hofner, B., Seibold, H., Vanschoren, J., & Bischl, B. (2017). OpenML: An R package to connect to the machine learning platform openML. *Computational Statistics, 34*, 1–15. https://doi.org/10.1007/s00180-017-0742-2.

Chi, J. T., Chi, E. C., & Baraniuk, R. G. (2016). k-pod: A method for k-means clustering of missing data. *The American Statistician*, *70*(1), 91–99. https://doi.org/10.1080/00031305.2015.1086685.

Contreras, P., & Murtagh, F. (2015). Hierarchical clustering. In C. Hennig, M. Meila, F. Murtagh, & R. Rocci (Eds.) *Handbook of Cluster Analysis,* (1st ed., Chap. 6, pp. 103–123). Chapman and Hall/CRC. https://doi.org/10.1201/b19706-6.

Dangl, R., & Leisch, F. (2020). Effects of resampling in determining the number of clusters in a data set. *Journal of Classification*, *37*(3), 558–583. https://doi.org/10.1007/s00357-019-09328-2.

Dinh, D. T., Huynh, V. N., & Sriboonchitta, S. (2021). Clustering mixed numerical and categorical data with missing values. *Information Sciences*, *571*, 418–442. https://doi.org/10.1016/j.ins.2021.04.076.

Eaton, J., & Haas, C. (1994). *Titanic: Triumph and tragedy.* Sutton Series. Patrick Stephens Ltd.

Fränti, P, & Sieranoja, S. (2019). How much can k-means be improved by using better initialization and repeats? *Pattern Recognition*, *93*(C), 95–112. https://doi.org/10.1016/j.patcog.2019.04.014.

Gionis, A., Mannila, H., & Tsaparas, P. (2005). Clustering aggregation. In *21st International Conference on Data Engineering (ICDE'05), IEEE Computer Society* (pp. 341–352). https://doi.org/10.1109/ICDE.2005.34.

Hennig, C. (2007). Cluster-wise assessment of cluster stability. *Computational Statistics and Data Analysis*, *25*, 258–271. https://doi.org/10.1023/A:1009769707641.

Hennig, C., & Liao, T. (2013). How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification. *Journal of the Royal Statistical Society - Series C (Applied Statistics)*, *62*(3), 309–369. https://doi.org/10.1111/j.1467-9876.2012.01066.x.

Hennig, C., Meila, M., Murtagh, F., & Rocci, R. (2015). *Handbook of cluster analysis*. New York: Chapman and Hall/CRC. https://doi.org/10.1201/b19706.

Huang, Z. (1998). Extension to the k-means algorithmus for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, *2*(6), 283–304. https://doi.org/10.1023/A:1009769707641.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification, 2*, 193–218. https://doi.org/10.1007/BF01908075.

Imbert, A., & Vialaneix, N. (2018). Décrire, prendre en compte, imputer et évaluer les valeurs manquantes dans les études statistiques - une revue des approches existantes. *Journal de la Société Francaise de Statistique*, *159*(2), 1–55.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, *31*(8), 651–666. https://doi.org/10.1016/j.patrec.2009.09.011.

Jimeno, J., Roy, M., & Tortora, C. (2021). Clustering mixed-type data: a benchmark study on kamila and k-prototypes. In T. Chadjipadelis, B. Lausen, A. Markos, T. R. Lee, A. Montanari, & R. Nugent (Eds.) *Data Analysis and Rationality in a Complex World* (pp. 83–91). Springer International Publishing. https://doi.org/10.1007/978-3-030-60104-1_10.

Jones, M. (1976). *Destination America*, 2nd edn. London: Weidenfeld & Nicolson.

Lange, K. (2013). *Optimization*. New York: Springer. https://doi.org/10.1007/978-1-4614-5838-8.

Leisch, F. (1999). Bagged clustering. *Working Papers SFB "Adaptive Information Systems and Modelling in Economics and Management Science", 51.* Vienna, Austria.

Little, R. J. A. (1988). Missing-data adjustments in large surveys. *Journal of Business & Economic Statistics*, *6*(3), 287–296. https://doi.org/10.1080/07350015.1988.10509663.

Little, R. J. A., & Rubin, D. B. (2019). *Statistical analysis with missing data* (3rd edn). John Wiley & Sons Inc. https://doi.org/10.1002/9781119482260.

Peña, J., Lozano, J., & Larrañaga, P. (1999). An empirical comparison of four initialization methods for the k-means algorithm. *Pattern Recognition Letters*, *20*(10), 1027–1040. https://doi.org/10.1016/S0167-8655(99)00069-0.

R Core Team. (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.

Rubin, D. B. (1987). Multiple imputation for nonresponse in surveys. John Wiley & Sons Inc. https://doi.org/10.1002/9780470316696.

Szepannek, G. (2018). clustMixType: User-friendly clustering of mixed-type data in R. *The R Journal*, *10*(2), 200–208. https://doi.org/10.32614/RJ-2018-048.

Szepannek, G., & Aschenbruck, R. (2021). *clustMixType: k-Prototypes Clustering for Mixed Variable-Type Data*. R package version 0.2-15.

Vanschoren, J., N van Rijn, J., Bischl, B., & Torgo, L. (2013). OpenML: Networked science in machine learning. *SIGKDD Explorations*, *15*, 49–60. https://doi.org/10.1145/2641190.2641198.

van Buuren, S. (2018). *Flexible imputation of missing data* (2nd edn). CRC Press.

van Buuren, S., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, *45*(3), 1–67. https://doi.org/10.18637/jss.v045.i03.

van Buuren, S., & Groothuis-Oudshoorn, K. (2021). *mice: Multivariate Imputation by Chained Equations*. R package version 3.13.0.

Vavrek, M. J. (2011). fossil: palaeoecological and palaeogeographical analysis tools. *Palaeontologia Electronica*, *14*(1), R package version 0.4.0.

van 't Veer, L. J., Dai, H., van de Vijver, M. J., He, Y. D., Hart, A. A. M., Mao, M., Peterse, H. L., van der Kooy, K., Marton, M. J., Witteveen, A. T., Schreiber, G. J., Kerkhoven, R. M., Roberts, C., Linsley, P. S., Bernards, R., & Friend, S. H. (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, *415*, 530–536. https://doi.org/10.1038/415530a.

Wagstaff, K. (2004). Clustering with missing values: No imputation required. In D. Banks, F. R. McMorris, P. Arabie, & W. Gaul (Eds.) *Classification, Clustering, and Data Mining Applications* (pp. 649–658). Springer. https://doi.org/10.1007/978-3-642-17103-1_61.

White, I. R., Daniel, R., & Royston, P. (2010). Avoiding bias due to perfect prediction in multiple imputation of incomplete categorical variables. *Computational Statistics & Data Analysis*, *54*(10), 2267–2275. https://doi.org/10.1016/j.csda.2010.04.005.

Yin, S., Gan, G., Valdez, E. A., & Vadiveloo, J. (2021). Applications of clustering with mixed type data in life insurance. *Risks, 9*(3), 47. https://doi.org/10.3390/risks9030047.