Applied Intelligence (2023) 53:7439–7454
https://doi.org/10.1007/s10489-022-03887-5

A partial order framework for incomplete data clustering

Hamdi Yahyaoui[1] · **Hosam AboElfotoh**[1] · **Yanjun Shu**[2]

Accepted: 11 June 2022 / Published online: 2 August 2022

Abstract

We propose in this paper a partial order framework for clustering incomplete data. The paramount feature of this framework is that it spans over a partial order that can be leveraged to establish data similarity. We present the underlying theoretical foundations and study the convergence of clustering algorithms in this framework. In addition, we present a partial order-based clustering algorithm (POK-means) that illustrates the embedding of K-means clustering algorithm in our framework. The first contribution of our method is that unlike methods based on imputation of the missing values, our method does not make any assumptions about missing data. Another important contribution is that it alleviates false dismissals caused by other interval-based similarity measures. The experimental results show that, although our method do not assume any prior knowledge of (or assumptions about) missing data, it is competitive to most of published incomplete data clustering methods that are based on assumptions about input data or imputation (e.g. methods based on partial or interval kernel distances) in accuracy and performance.

Keywords Partial order · Lattice · Clustering · Incomplete data · Lower bounding

1 Introduction

Clustering is one of the common techniques that help in discovering the structure of data and recognizing its inherent patterns. By grouping similar objects in the same cluster and keeping different (dissimilar) objects in separate clusters, clustering paves the way for further data analyses and interpretation. Data clustering has many important applications, e.g., image analysis, bioinformatics, market analysis, etc. More insights on clustering can be found in [5, 19].

The input to a clustering algorithm is a collection of n d-dimensional objects (vectors) $X = \{X_1, X_2, \ldots, X_n\}$ and for some algorithms an integer K, which denotes the desired number of clusters. Each X_i is a set $\{X_{i1}, X_{i2}, \ldots, X_{id}\}$, where X_{ij} represents the j^{th} attribute (feature) of the i^{th} object. The algorithm groups the objects into K groups (subsets) such that the dissimilarity of objects within each group is minimized. For instance, some clustering algorithms try to minimize the Sum of Squared Errors (SSE), which is a measure of cluster cohesion. SSE is defined as follows:

$$SSE = \sum_{i=1}^{K} \sum_{o \in C_i} d(o, \mu_i)^2 \tag{1}$$

Where C_i is a cluster among K clusters, μ_i is the *centroid* (or the *center*) of C_i and $d(o, \mu_i)$ is the distance between an object o and μ_i.

The most commonly distance function that is used in clustering is the Euclidean Distance (EU) and is defined as:

$$EU(X, Y)^2 = ||X - Y||_2 \tag{2}$$

Data can be incomplete, i.e., includes missing data. Several reasons can be behind this such as measurement device malfunctioning, or even the absence of a value. This creates a certain uncertainty that should be taken into account while establishing the similarity between data objects.

✉ Hamdi Yahyaoui
hamdi@cs.ku.edu.kw

Hosam AboElfotoh
hosam@cs.ku.edu.kw

Yanjun Shu
yjshu@hit.edu.cn

[1] Computer Science Department, Kuwait University
Safat 13060, State of Kuwait

[2] School of Computer Science and Technology, Harbin Institute of Technology Harbin, China

Many methods based on *imputation* [2, 3, 9, 24, 27] were proposed to handle incomplete data. Imputation means filling in missing data based on some assumptions about data probability distribution or on a prior knowledge of the data structure. Methods based on imputation by specific values may provide biased classification/clustering results and high misclassification rate.

Other methods rely on interval-based distances to deal with incomplete data. These methods represent the distance as a single value. We show in the next section that they don't satisfy the lower bounding constraint, which requires that the distance between these intervals is lower than or equal to the Euclidean distance between the corresponding data objects (the maximum possible Euclidean distance in case of missing values). Not satisfying this constraint may result in false dismissals, where some objects are falsely dismissed from their right clusters. This shortcoming is mainly due to a "flattening" process that is followed by these methods while establishing the distance between such objects. Accordingly, the main motivation of our work is to develop a method applicable to incomplete data while satisfying the lower bounding constraint.

We propose in this paper a Partial Order Framework (POF) for clustering incomplete data. Data similarity is captured by an interval that approximates the distance between data objects. A partial order is then built on top of these intervals to compare these approximations. Our method alleviates false dismissals that may be caused by commonly used distance measures such as those proposed in [11, 12].

To summarize, the main contributions of this paper include:

- The proposal of a partial order framework for clustering incomplete data without any assumption on the data.
- The proposal of a distance for incomplete data that satisfies the lower bounding constraint and so minimizes false dismissals.
- The study of convergence for clustering algorithms that can be embedded in the proposed framework.
- The specification and the implementation of the K-means clustering algorithm (POK-means) in the elaborated framework to deal with incomplete data together with experimental results that show that POK-means is competitive to other related published clustering methods for incomplete data.

The remainder of the paper is organized as follows. Section 2 is devoted to the related work and shortcomings in existing methods. Section 3 presents the partial order clustering framework and POK-means algorithm for clustering incomplete data. Section 4 includes the experimental results. Finally, we discuss possible enhancements of our method in Section 5.

2 Related work

Partial order clustering was applied to solve different issues related to systems verification, user preferences clustering and chains clustering. In [1], a partial order framework is used to mitigate the effects of the state-space explosion problem by clustering the state-space of concurrent processes. The problem of clustering a set of chains into k clusters is considered in [26]. Clustering of chains can be used in applications, such as user preference surveys, decision analysis, voting systems, and bioinformatics. The author in [17] provides a comparison of three approaches of clustering totally and partially ordered subsets of a set of items. In [25], a partial clustering algorithm (not every point is included in a cluster) is introduced for the sake of minimizing the effect of outlier and noise points. A cluster-similarity metric that is based on partial order was developed for rooted phylogenetic trees in [8]. Phylogenetic trees are used to describe relations among species. Despite these insights, to the best of our knowledge partial order clustering has not been applied to handling incomplete data before. Our proposed algorithm is the first incomplete data clustering algorithm that is based on a partial order framework.

Uncertainties may arise while assessing incomplete data similarity. Some feature values or readings may be missing, e.g., $X_i = \{20, 35, \mathbf{?}, 17, 14\}$, or values are only known to belong to a certain range. To deal with such uncertainty, many methods were proposed in the literature [9, 14, 22]. Among these we mention: listwise deletion, pairwise deletion, mean substitution, last observation carried forward, regression imputation, maximum likelihood, and multiple imputation. The Optimal Completion Strategy (OCS) [7] relies on imputation of missing data based on some optimal methods to determine better estimates. However, these solutions either have assumptions about the data distribution such as regression imputation [10] or they require extensive computation such as multiple imputation [21]. Therefore, they are likely to produce biased results if there is no prior knowledge about missing data.

Other methods, which are not based on imputation, were devised to deal with missing data. Among these methods the one that spans over a Nearest Prototype Strategy (NPS) [7] that replaces missing values with the corresponding attribute values of the nearest prototype while clustering. Another appealing method in this category is the fuzzy-based clustering [7, 18], which leverages the concept of fuzziness to cluster incomplete data. In such a setting, an object may belong to more than one cluster with different probabilities. This can be expressed using the fuzzy membership concept. To deal with incomplete data, two commonly used strategies [7] were proposed:

- Whole Data Strategy (WDS): consists in deleting all incomplete data vectors if they constitute a small fraction of the data. Thus, WDS cannot be used when a significant fraction of the data is missing.
- Partial Distance Strategy (PDS): relies on only existing features (components) to compute the distance between two vectors. The distance is scaled by the proportion of used components. The limitation of PDS is that it cannot be applied when for example missing values exist alternatively in both vectors.

These strategies may introduce a biased similarity measurement that can be carried over in all the iterations of the clustering algorithm by ignoring a part of the data or deleting it completely.

Fuzzy C-means was also extended in [11] to deal with incomplete data based on Nearest-Neighbor Intervals (NNI) strategy. NNI uses an interval-based distance function to assess the similarity between data objects. It represents a missing data X_i by the minimum and maximum of the neighbors' corresponding attributes, i.e., $X_i = \{[X_i^-, X_i^+]$, where X_i^- and X_i^+ are the lower and upper bound values, respectively. NNI comes with an Interval-based Distance (*ID*) and relies on a total order in assessing the similarity between data objects. The distance $ID(X, Y)$ between two samples $X = \{[X_1^-, X_1^+], \ldots, [X_d^-, X_d^+]\}$ and $Y = \{[Y_1^-, Y_1^+], \ldots, [Y_d^-, Y_d^+]\}$ is defined as:

$$ID(X, Y)^2 = \sum_{i=1}^{d} |X_i^+ - Y_i^+|^2 + \sum_{i=1}^{d} |X_i^- - Y_i^-|^2 \quad (3)$$

Unfortunately, the distance *ID* does not satisfy the lower bounding constraint (as we show in the following Claim) and subsequently may result in false dismissals. In the context of clustering algorithms, a false dismissal means a decision taken by the algorithm that an object does not belong to its proper cluster. This may lead to inaccurate clustering results or wrong query matchmaking. More discussion about false dismissals can be found in [6, 13, 20].

Claim 1: Let Y be a known feature and X be a missing (unknown) value = $[X^-, X^+]$. Then, $ID(X, Y) >= EU(X, Y)$.

Proof We prove the claim for one-dimensional vectors. However, it can be generalized to multi-dimensional case. □

Suppose X and Y are points on the line -∞ to +∞ and let X be the actual value.

$ID(X, Y)^2 = (Y - X^-)^2 + (Y - X^+)^2, EU(X, Y)^2 = (Y - X)^2$

Case 1: $X^- < X < X^+ < Y : Y - X^- > Y - X.$
Case 2: $X^- < X < Y < X^+ : Y - X^- > Y - X.$
Case 3: $X^- < Y < X < X^+ : X^+ - Y > X - Y.$
Case 4: $Y < X^- < X < X^+ : X^+ - Y > X - Y.$

Since $ID(X, Y)^2 = (Y - X^-)^2 + (Y - X^+)^2 > \max((Y - X^-)^2, (Y - X^+)^2) > (Y - X)^2$, then in all cases $ID(X, Y)^2 > EU(X, Y)^2$.

Based on Claim 1, the *ID* distance does not satisfy the lower bounding constraint required in computing distances or dissimilarity measures since it may return values that are higher than the maximum possible Euclidean distance. Therefore, it can lead to false dismissals.

An Interval Kernel Fuzzy C-means method (IKFCM) is proposed in [12] as an improvement of previous published fuzzy methods such as [11]. In this method, the data is mapped into a higher dimensional feature space using a Gaussian function. IKFCM uses a nonlinear kernel-induced distance to replace the Euclidean distance. The authors provide also an application of their method on OCS, NPS, WDS and PDS to produce KOCS, KNPS, KWDS and KPDS as kernel-based methods. The proposed interval kernel distance $IKD(X, Y)$ for two intervals X and Y is:

$$IKD(X, Y) = \sqrt{2 - 2\exp\left(-\frac{(X^- - Y^-)^T(X^- - Y^-) + (X^+ - Y^+)^T(X^+ - Y^+)}{2 \times \sigma^2}\right)} \quad (4)$$

The interval kernel distance is used for the sake of increasing the separability between the data and consequently improving the clustering quality. However, There is no proof in [12] that the proposed distance does not result in false dismissals.

Once again, we can show (Claim 2) that based on Claim 1, the actual kernel distance between two objects may be less than the computed interval kernel distance between these objects in IKFCM. Therefore, false dismissals may occur based on interval kernel distance clustering.

Claim 2: Let Y be a known feature and X be a missing (unknown) value = $[X^-, X^+]$. Let X_a be the actual value of X.

$$IKD(X, Y) = \sqrt{2 - 2\exp\left(-\frac{(X^- - Y)^2) + (X^+ - Y)^2)}{2 \times \sigma^2}\right)} \quad (5)$$

Let $KEU_a(X, Y)$ be the actual kernel distance (kernel mapping of the Euclidean distance) computed using the same function.

$$KEU_a(X, Y) = \sqrt{2 - 2\exp\left(-\frac{(X_a - Y)^2}{2 \times \sigma^2}\right)} \quad (6)$$

Then, $IKD(X, Y) >= KEU_a(X, Y)$.

Proof

$$IKD(X, Y) = \sqrt{2 - 2\exp\left(-\frac{ID(X, Y)^2}{2 \times \sigma^2}\right)} \quad (7)$$

$$KEU_a(X, Y) = \sqrt{2 - 2\exp\left(-\frac{EU(X, Y)^2}{2 \times \sigma^2}\right)} \quad (8)$$

Since $ID(X, Y)^2 > EU(X, Y)^2$ (Follows from Claim 1), $IKD(X, Y) > KEU_a(X, Y)$. □

Recently, imputation-based methods have been used together with objective functions optimization to handle missing data in [24]. More precisely, the authors propose a novel K-means based clustering algorithm for incomplete data (KMID), which unifies the clustering and imputation into one single objective function. Furthermore, they design an alternate optimization algorithm to solve the resultant optimization problem and theoretically prove its convergence. This approach inherits imputation methods shortcomings such as introducing of bias at the data level.

In [23], the authors propose the KM-IMI algorithm, which uses the method of adding weights and analyzing perturbation distance of cluster centroid to cluster incomplete datasets. The k-means algorithm is first applied to the subset of data with non-missing values. Then, the weights are used to find the optimal imputation that leads to the best possible clustering result. Finally, a partition clustering algorithm is used to get the final clustering result.

In summary, imputation-based methods have some assumptions on the data in order to derive and replace missing data, which restrict their applicability in some cases while fuzzy based methods rely on a total order relation to assess data similarity in classical clustering algorithms, which is the main reason behind violating the lower bounding constraint. Indeed, interval-based or missing data is captured by intervals, but this range of possibilities is lost while doing a kind of "flattening" of these intervals to compute the distance between data objects. Based on these observations, we initiated a research that aims to devise a new framework for establishing data similarity, which should take into account the limitations of the existing methods.

We follow in this work a different method in which we take into consideration missing data and represent the distance uncertainty caused by such missing as an interval. Such interval represents the range of possible distances between two data instances. Furthermore, we show that this method provides a basis for a general *framework* for handling both cases of complete and incomplete data, with real or discrete-valued features.

3 Partial order based clustering

In this section, we present theoretical foundations underlying our proposed partial order clustering framework and an embedding of the K-means clustering algorithm in it. It is worth to mention that the proofs of the claims in this section are provided in the Appendix A.

3.1 Preliminaries and definitions

Before we discuss the details related to the proposed partial order-based clustering, we provide some useful definitions that help in grasping the underpinnings of our framework.

A partially order set (poset) consists of a set with a binary relation that establishes an order between two elements of that set. The relation is partial, which means that two elements may be non comparable. A partial order relation has three properties: It is reflexive, anti-symmetric, and transitive.

A partial order in which every pair of elements is comparable is called a total order or a chain. Hence, a total order is a special case of a partial order relation. A dual notion of a chain is antichain, which is a subset of a poset in which no two distinct elements are comparable.

The partial order relation comes with upper and lower bound concepts for subsets of posets. An upper bound of a subset X of a poset P is an element of P, which is greater than or equal to every element of X. Dually, a lower bound of a subset X of a poset P is an element of P, which is less than or equal to every element of X. The supremum (or Least Upper Bound) of a subset X of a poset P is the lowest element in P that is greater than or equal to all the elements of X. Dually, the infimum (or Greatest Lower Bound) of a subset X of a poset P is the greatest element in P that is less than or equal to all the elements of X.

A poset $(L, \preceq)$ is a complete lattice if every subset A of L has both a greatest lower bound and a least upper (also called the meet) bound (also called the join) in L. We show in this paper how the complete lattice concept can be leveraged in guaranteeing the convergence of our proposed clustering algorithm.

3.2 A partial order framework for data clustering

The proposed Partial Order Framework (POF) for data clustering relies on relaxing the total order, generally used while establishing the similarity between data objects, and replacing it with a partial order. This order reflects the uncertainty that may arise in handling incomplete data.

POF can be applied either directly on the data domain $\mathcal{D}$ by defining a similarity measure $\mathcal{I}$ on $\mathcal{D} \times \mathcal{D}$. Its co-domain is a partially order set $\mathcal{P}$. POF can also be applied on an abstract domain $\mathcal{A}$ that is the result of an abstraction operation of the domain $\mathcal{D}$ using a mapping $\mathcal{M}$. In this case, the similarity measure $\mathcal{I}'$ is defined on the elements of $\mathcal{A} \times \mathcal{A}$ and its co-domain is a partially ordered set $\mathcal{P}'$. Figure 1 outlines these two modus operandi of POF. It is worth noting that $\mathcal{M}^*$ is the extension of $\mathcal{M}$ to $\mathcal{D} \times \mathcal{D}$.

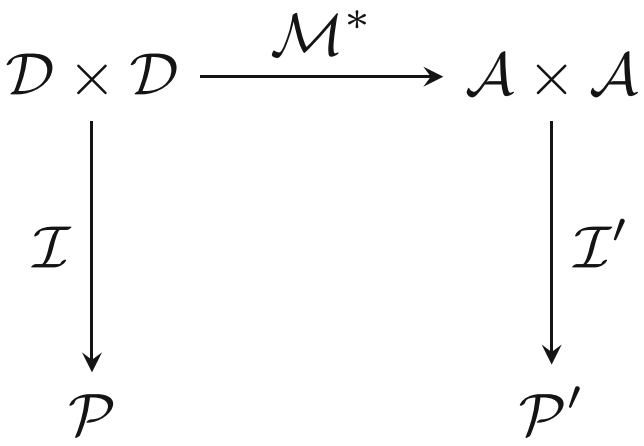


Fig. 1 Partial order modus operandi

3.2.1 Example

As a concrete example, POF can for instance handle directly interval data (the domain D is the interval data in this case) by defining a relation $\mathcal{I}$ that can be an interval-based distance between data objects. Such a distance can rely on a partial order that is defined on intervals. The distance results can be for instance intervals that are equipped with a partial order in a poset P.

Also, POF can operate on symbolic sequences (so the domain A in this case is a set of symbolic sequences) that are derived from raw data which can be time series (the domain D in this case is a set of time series) by defining a similarity relation $\mathcal{I}'$ in a complete partial order. Symbolic sequences can be derived from raw data using a transformation (mapping $\mathcal{M}$) like Symbolic Aggregate approXimation (SAX) [13].

3.3 Partial order on intervals

In this section, we provide an example of a partial order that can be defined in POF. More precisely, we propose the following partial order $\preceq$ on the space of intervals $\mathcal{L}_{\mathcal{I}}$.

$$[a_L, a_R] \preceq [a'_L, a'_R] \text{ iff } a_L \leq a'_L \wedge a_R \leq a'_R \tag{9}$$

Proposition 1 *$\preceq$ is a partial order.*

Figure 2 outlines the proposed partial order relations between intervals in a Hasse diagram. The bottom element is the interval [min(α_i),min(β_i)], which denotes the element that is dominated, with respect to the partial order $\preceq$, by all intervals while the top element, which is [max(α_i), max(β_i)], denotes an interval that dominates all the intervals in lower levels.

Proposition 2 *($\mathcal{L}_{\mathcal{I}}$, $\preceq$) is a complete lattice.*

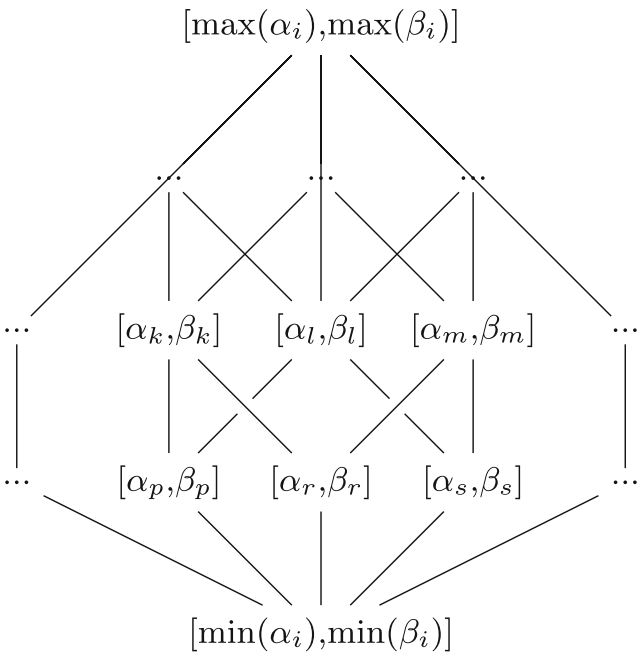


Fig. 2 Interval distance lattice

It is worth to mention that this partial order is used to compare interval distances and derive possible minimum distances with respect to the partial order while deciding about cluster membership.

3.4 Data representation and similarity in POF

POF is a general framework and can cope with incomplete data and also other data types. We explain in what follows the representation of different data types in POF together with the specification of an interval-based distance that can be applied on such data. More precisely, the distance between two data objects is represented as an interval $[L, U]$, where L and U denote respectively the minimum and maximum possible distance values between these two data objects. We discuss in the sequel the representation of different data types in POF and the calculation of the interval distance between two data objects:

– Interval data: an interval data x is represented by the interval [*xmin*,*xmax*]. The interval distance in POF, $POD(x, y)$, between two interval data objects x and y is the interval [$d_{\min}(x, y)$,$d_{\max}(x, y)$], where:

$$d_{\min}(x, y) = \begin{cases} 0, & \text{if } x \text{ and } y \text{ overlap;} \\ \min(EU(xmax, ymin), EU(ymax, xmin)) & \text{Otherwise.} \end{cases}$$

$$d_{\max}(x, y) = \max(EU(xmax, ymin), EU(ymax, xmin))$$

Example: Let us consider two intervals A = [1,3] and B = [2,4] that overlap. The interval distance between A and B in POF is [0,3]. Now, let us consider two intervals

A = [1,2] and B = [3,4] that do not overlap. The interval distance between C and D in POF is [1,3].

- Numerical and categorical data: any numerical or categorical (complete) data x can be represented by the interval [*xmin*,*xmax*] with *xmin*=*xmax*=x. The interval-based distance in POF, $POD(x, y)$, between two data objects x and y is the interval $[d_{\min}(x, y), d_{\max}(x, y)]$ with $d_{\min}(x, y) = d_{\max}(x, y)$. Here, the partial order between the intervals is a total order since all the intervals are comparable in this case. So classical clustering algorithms can be embedded in the framework in a straightforward manner.
- Multivariate data: A multivariate data x is represented by a vector $< x_1, x_2, \ldots, x_n >$, where x_i is the data representation of the i^{th} feature in POF. The interval-based weighted distance $POWD(x, y)$ between two multivariate data objects x and y is:

$$POWD(x, y) = \left[\sum_{i=1}^{n} w_i d_{\min}(x_i, y_i), \sum_{i=1}^{n} w_i d_{\max}(x_i, y_i)\right]$$

where w_i is the weight of the i^{th} feature and $\sum_{i=1}^{n} w_i = 1$. This weight reflects the importance of an attribute and can be derived automatically for example from the correlations between features. The sum constraint guarantees that the weighted distance lower bounds the maximum possible value of the Euclidean distance between x and y as proved later in this paper.

- Incomplete data: an incomplete data i can be represented by i = [*xmin*,*xmax*] where *xmin* and *xmax* can respectively represent minimum and maximum values of nearest neighbors of such data. Then, the distance between two incomplete data is computed as discussed in the interval data case.

3.5 Distance approximation issues in the related work

We show in this section how our proposed interval-based distance *POD* can solve the issues that come with the published distances for handling incomplete data. Let us consider the following data vectors: $T_1 =< 1, 4, 1, 2 >$, $T_2 =<?, 3, 1, 3 >$, $T_3 =< 6, 5, 4, 1 >$, $T_4 =< 1, 4, 1, 1 >$, and $T_5 =< 2, 4, 2, 2 >$. The partial distance between T_1 and T_2 is:

$$PDS(T_1, T_2) = \sqrt{\frac{4}{3}(1 + 0 + 1)} = 1.63$$

Based on the partial distances between T_2 and other points, the best three nearest neighbors to T_2 are T_1, T_4 and T_5. So, the missing value "?" is replaced by the interval [1,2]. The distance computations are as follows:

$$\begin{aligned}
ID(T_1, T_2) &= \sqrt{(0 + 1 + 0 + 1) + (1 + 1 + 0 + 1)} = \sqrt{5} = 2.24\\
POD(T_1, T_2) &= [\sqrt{0 + 1 + 0 + 1}, \sqrt{1 + 1 + 0 + 1}] = [1.41, 1.73]\\
IKD(T_1, T_2) &= \sqrt{2 - 2 * \exp(-\frac{5}{2 \times 0.75^2})} = 1.41, \text{with } \sigma^2 = 0.75 \text{ as suggested in (12)}\\
KEU(T_1, T_2) &\in \left[\sqrt{2 - 2 * \exp(-\frac{2}{2 \times 0.75^2})}, \sqrt{2 - 2 * \exp(-\frac{3}{2 \times 0.75^2})}\right] = [1.29, 1.36]
\end{aligned}$$

ID in this case violates the lower bounding constraint since the maximum Euclidean distance is 1.73. For *IKD*, the computed distance is lager than the maximum possible kernel mapping of the actual Euclidean distance *KEU*, which is 1.36.

Theorem 1 *Any approximation based on the weighted distance POWD satisfies the lower bounding constraint.*

3.6 Specification of K-means clustering algorithm in POF

K-means is partitional clustering algorithm that tries to minimize SSE. K-means does not require that a cluster center belongs to the cluster. When specified in POF, any clustering algorithm that minimizes SSE should be updated to minimize an interval [SSE-Min, SSE-Max], with respect to the order $\preceq$, that captures the sum of the distances between data points and the cluster center points. In case of absence of incomplete data or interval data (complete data case), SSE-Min coincides with SSE-Max and the specification of such algorithm in POF is equivalent to its classical implementation.

Algorithm 1 outlines the specification of the K-means clustering algorithm in POF, which we call POK-means. It starts by replacing missing values (if any) by intervals having as extremities (Line 3): the minimum and the maximum of the interval extremities' values for the nearest neighbors. Then, it initializes the clusters' centers using any good seeding algorithm such as the one used in K-means++ (Line 4). The data interval center points can be used to apply such seeding algorithm. The clustering membership of a data element is determined based on the minimal interval distance with respect to the partial order $\preceq$ and the clusters' centers (Lines 5 to 10). In case of two or more incomparable intervals, the membership is determined randomly. After determining the membership of all data elements, the new centers are computed as the data intervals' averages (Line 11). The minimum and maximum values (extremities) of intervals' averages are respectively the averages of the minimum and maximum values of the averaged intervals. This clustering procedure is repeated until getting an SSE interval that does not improve the current solution, i.e., not better than the current SSE with respect to the order $\preceq$.

Algorithm 1 POK-means clustering algorithm.

input: k: number of clusters, R: number of nearest neighbors, $\mathcal{S}$: Set of data points
output: $\mathcal{C}$ Clusters
Replace the missing values in $\mathcal{S}$ using R nearest neighbors and the partial distance
Select the set $\mathcal{M}$ of initial k means from $\mathcal{S}$ and add them to $\mathcal{C}$
do
  for each $s \in \mathcal{S}$ **do**
    Compute interval distances $\mathcal{L}_{\mathcal{I}}$ between each of the cluster means in $\mathcal{M}$ and s using POWD
    Determine the element $m \in \mathcal{C}$ having the minimal distance in $\mathcal{L}_{\mathcal{I}}$ with respect to the order $\preceq$
    Assign s to the cluster defined by m and update $\mathcal{C}$
  end for
  recompute $\mathcal{M}$ as the set of new cluster means
  recompute SSE
while SSE is decreasing with respect to the order $\preceq$
return $\mathcal{C}$ clusters

The algorithm is minimizing SSE in each iteration by rejecting any SSE that is not better than the previous SSE in terms of the intervals partial order. We call such constraint *strict decreasing chain constraint*.

3.7 Complexity analysis and convergence

The computation of the intervals for nearest neighbors of missing data includes computing two values: minimum and maximum distances using the partial distance. This computation requires in the worst case $2 * n(n-1)d$ operations, where n is the data size and d is the number of features. Thus, for each alternating iteration among t iterations, the complexity of the preprocessing step is equal to $O(\frac{n(n-1)d}{t})$, which is the same as the on for IKFCM algorithm [12].

The complexity of the POK-means clustering step is $O(nkdt)$, where k is the number of clusters. Therefore, the time complexity for each alternating step is $O((\frac{n}{t}+k)nd)$, which is better than IKFCM that has a time complexity equal to $O((\frac{n}{t}+k^2)nd)$ as published in [12].

The strict decreasing chain constraint for SSE guarantees that POK-means converges. In what follows, we discuss the convergence for the strict version of the algorithm and also the version where such condition is relaxed.

3.7.1 Strict version

The following theorem proves the convergence of the strict version of POK-means.

Theorem 2 *POK-means, in its strict version, converges to a local minimum of the lattice* $(\mathcal{L}_{\mathcal{I}}, \preceq)$.

Figure 3 outlines an example of the possible visited elements in the lattice $\mathcal{L}_{\mathcal{I}}$ while executing POK-means. The path of the elements is a decreasing chain and it converges to an element (representing SSE) of an anti-chain without visiting any other element of the ant-chain. It is worth to mention that the strict version of POK-means guarantees fast convergence, but the quality may be not good.

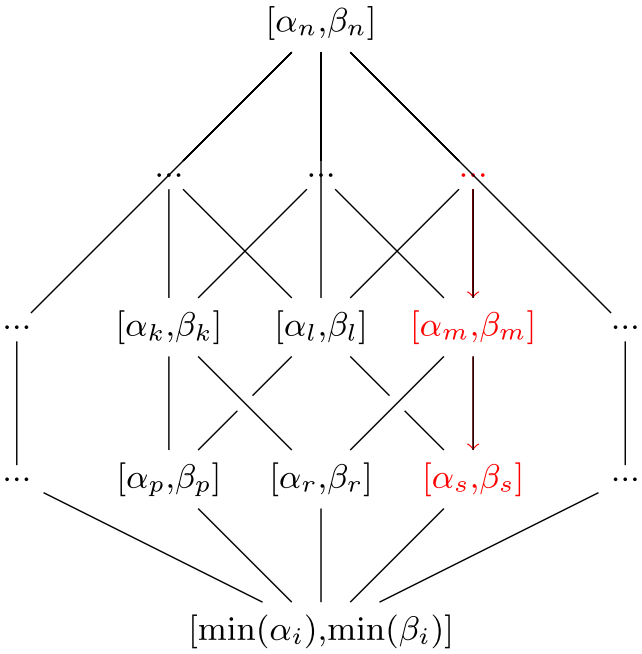


Fig. 3 Strict downward elements visit. The path of the decreasing chain is marked in red. This path does not include incomparable elements and may end up with a bad local minimum

3.7.2 Relaxed version

To improve the quality of the obtained solution, the decreasing chain constraint is relaxed and SSE intervals that are not better (lower in order) than previous SSE intervals are accepted. This allows the algorithm to visit anti-chain elements as opposed to the strict version and so may end up with a better clustering results but with a certain time cost compared to the strict version. Figure 4 outlines possible visited elements in the lattice $\mathcal{L}_{\mathcal{I}}$.

4 Experiments

4.1 Experiments setting

We conduct experiments in this paper to show the effectiveness of our framework in clustering incomplete data. More precisely, we show how the proposed distance minimizes dismissals by having acceptable rates of misclassifications. Our experiments are done on known datasets [4, 16], which are commonly used in clustering incomplete data [11, 12, 18]. These datasets are: Iris, Wine, Wholesale,

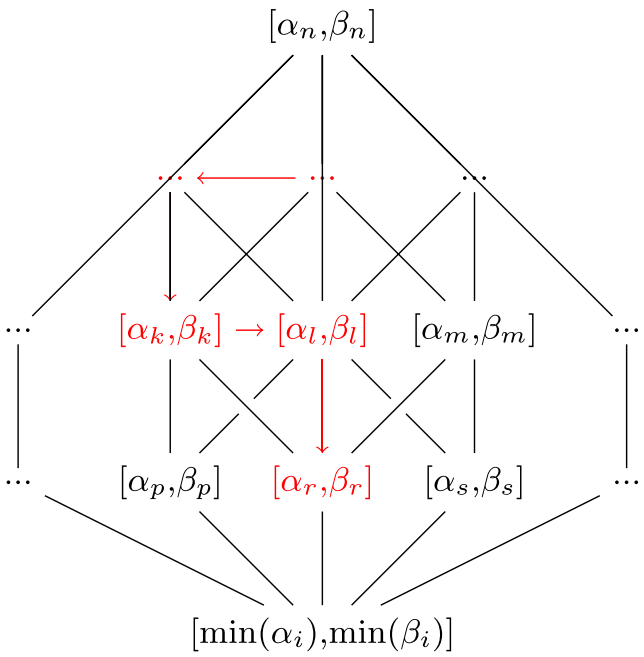


Fig. 4 Relaxed elements visit. The path of the visited elements is marked in red. This path may include incomparable elements and ends up with a solution that is better than the strict one but takes more time to find a good clustering configuration

Breast Cancer and Thyroid Gland. The descriptions of the three datasets are as follows:

- Iris dataset: a multivariate dataset of Iris flowers. It consists of a total of 150 samples from three species (clusters): Iris Setosa, Iris Virginica and Iris Versicolor. It has 4 attributes (features) related to the length and the width of the sepals and petals.
- Wine dataset includes 178 features that are the obtained from the analysis of wines grown in the same region but from different cultivars. The dataset includes three classes in which the first one contains 59 instances, the second has 71 instances and the third includes 48 instances.
- Wholesale Customers dataset includes 6 features and can be split into 2 categories according to the Channel index. There are 298 instances in the first category and 142 instances in the second one.
- Breast cancer dataset: a multivariate dataset that includes 699 instances regarding breast cancer. It has 9 attributes related to characteristics of the cell nuclei present in a breast mass image. The dataset can be divided into malignant and benign clusters.
- Thyroid gland dataset: it has three classes: normal with 150 instances, hyper with 35 instances and hypo with 30 instances. It includes five attributes: T3-resin uptake test (T3), Total Serum thyroxin (TTS), Total serum triiodothyronine (TST), basal thyroid-stimulating hormone (TSH) and Maximal absolute difference of TSH value after injection of 200 micrograms of thyrotropin-releasing hormone (DTSH).

The missing values' positions are selected randomly and they represent a certain rate of the whole data. This rate is either: 5%, 10%, 15%, or 20% as used in previous works [12]. There are two constraints that should be satisfied while including missing data, which consist in having at least one non-missing value per attribute and per instance. Missing data is replaced by the minimum and maximum values of N nearest neighbors, where N is a parameter that can be tuned as proposed in [12]. We compare our results with those published in [12], which include the following published methods: WDS, PDS, OCS, NPS [7], NNI [11], KWDS, KPDS, KOCS, KNPS and IKFCM [12], KMID, KNNF, EMF, MF, ZF [24] and KM-IMI [23]. In KNNF (KNN-Filling), the missing values are filled with the mean feature of the K-nearest neighbors. In EM (Expectation Maximum), the algorithm estimates the model parameters for filling incomplete data. In MF (Mean Filling) the algorithm fills the missing values with mean values, as introduced in the related work and in ZF (Zero Filling), the algorithm firstly standardizes the data matrix and imputes zeros on the missing values.

The relaxed version of POK-means is applied on the aforementioned three datasets with different missing rates. The weighted distance for multivariate data is leveraged as presented in Section 3.4. The relaxed version of the algorithm is implemented and assessed according to two commonly used metrics:

- Mean number of misclassifications: the mean of 10 experiments' results as done in [12] regarding the clustering errors compared to the ground truths in the original datasets.
- Mean number of iterations to termination: the mean of 10 experiments' results regarding the number of iterations required by POK-means to converge.

The initialization of cluster centers is important so POK-means is using a non-random selection of these centers. The current POK-means relies on an adaptation of the K-means++ initialization version to intervals as explained in Section 3.6. However, POK-means is open to any other centers selection procedure. The current implementation also calculates weights for attributes and uses them in the attributes' distances summation. These weights can be either tuned or derived automatically as will be explained in what follows. To derive a weight for each attribute, the attribute pairwise correlation matrix (can be pairwise covariance matrix for other datasets) is extracted and each attribute weight is calculated as the sum of the attribute

row divided by the sum of the all the matrix elements. This weight reflects to which extent the other attributes depend on that attribute. More formally, let $Corr$ be the pairwise covariance matrix for d attributes of the dataset:

$$Corr = \begin{bmatrix} corr_{11} & corr_{12} & \dots & corr_{1d} \\ corr_{21} & corr_{22} & \dots & corr_{2d} \\ \dots & \dots & \dots & \dots \\ corr_{d1} & corr_{d2} & \dots & corr_{dd} \end{bmatrix}$$

The weight w_i of an attribute i is provided in (10).

$$w_i = \left| \frac{\sum_{j=1}^{d} corr_{ij}}{\sum_{i,j=1}^{d} corr_{ij}} \right| \tag{10}$$

w_i reflects to which extent other features depend on the i^{th} feature. So the weighted distance can help in increasing the separability between cluster elements at the clustering step. This will be shown empirically in what follows.

4.2 POK-means clustering evaluation

The experimental results are provided in Tables 1, 2, 3 and 4. Best results are marked in bold. The results show that:

- POK-means enjoys good accuracy results compared to other techniques: kernel based clustering techniques (WDS, PDS, OCS, NPS [7], NNI [11], KWDS, KPDS, KOCS, KNPS and IKFCM [12]), and imputation based techniques (KMF, KNNF, EMF, MF, ZF [24] and KMIMI [23]) and comparable results to KMID [24]. It is generally better than the Kernel based techniques and competitive to the imputation based techniques. The weighted distance and the relaxed convergence strategies are the main ingredients to achieve good results in POF. More precisely, the weighted distance leads to a more precise clustering since it gives higher weight to the attributes that can discriminate well objects while clustering and consequently a better separability between data. On the other hand, the relaxed convergence strategy avoids bad local minimum solutions while visiting the lattice elements. However, it does not guarantee obtaining the optimal clustering solution.
- POK-means has a fast convergence compared to the kernel based techniques and close performance results to imputation based techniques. This can be explained by the fact that the imputation allows the clustering algorithm to converge in fewer steps. However, this comes with a stability issue as the standard deviation values of the imputation based clustering results indicate in the sequel.

Compared to Kernel based clustering techniques, the results for the Iris dataset show an improvement in terms of the mean number of misclassifications that is between 14.84% and 51.71%. Compared to imputation based clustering techniques, POK-means has better results than all imputation based techniques except with ZF. The

Table 1 Accuracy and performance experimental results for incomplete Iris dataset

Rate	WDS	PDS	OCS	NPS	NNI	KWDS	KPDS	KOCS	KNPS	IKFCM	KMID	KNNF	EMF	MF	ZF	POK-means
Mean number of misclassifications																
0	16	16	16	16	16	16	16	16	16	16	32.9	16.6	16.6	16.6	**6**	11
5	17.2	17.2	17.7	17.5	17.1	16.7	16.7	16.3	16.1	16.3	35.9	30.1	27.4	30.1	**8.4**	11.9
10	16.1	16.8	16.5	16.9	16.0	15.8	16.1	16.4	15.7	15.5	25	39.6	21.8	22.6	13.9	**12.1**
15	17.2	17.4	17.8	17.1	15.6	17.1	16.3	16.4	15.7	15.5	34.6	44.6	16.1	24.4	16.3	**13.2**
20	17.8	17.4	17.2	17.2	16.9	23.4	16.7	16.8	16.6	16.1	39	49	25.2	34.8	27.4	**11.3**
Mean number of iterations to termination																
0	27.9	27.9	27.9	27.9	27.9	31.7	31.7	31.7	31.7	31.7	6	6.9	6.9	6.9	4.7	13
5	29.6	28.8	37.2	30.0	29.6	33.5	32.8	50.5	33.4	33.2	6.6	6.6	5.8	4.9	4.9	13.3
10	29.8	27.2	45.0	30.0	25.6	33.3	29.7	40.8	32.5	30.0	10	6.6	7.1	6.4	7.6	12.3
15	29.6	28.3	66.3	31.6	26.7	32.7	29.9	49.2	34.3	29.7	8.7	4.9	4.4	5.2	5	11.4
20	36.0	30.3	48.9	34.7	26.0	44.3	32.5	49.1	38.7	28.8	8.4	7.8	4.5	6.6	4.3	10.4
Standard deviation of misclassifications																
0	0	0	0	0	0	0	0	0	0	0	26.99	0.51	0.51	0.51	0	0
5	1.14	0.92	1.34	1.08	1.37	1.57	1.16	1.16	1.10	1.16	25.95	17.36	23.01	23.46	1.43	1.66
10	2.23	1.32	1.08	1.29	2.11	2.04	1.10	1.41	1.49	1.65	13.16	17.43	18.03	4.37	3.63	1.91
15	1.69	1.26	1.48	1.52	2.17	1.60	1.49	1.43	1.42	1.84	14.50	17.45	1.85	2.17	4.73	2.20
20	2.15	2.01	2.62	1.99	1.85	18.31	2.50	2.49	2.22	1.52	15.89	16.28	20.76	15.88	18.21	2.11

Table 2 Accuracy and performance experimental results for incomplete Wine dataset

Rate	WDS	PDS	OCS	NPS	NNI	KWDS	KPDS	KOCS	KNPS	IKFCM	KMID	KNNF	EMF	MF	ZF	POK-means
Mean number of misclassifications																
0	9	9	9	9	9	8	8	8	8	8	3	3	**3**	3	3.4	9
5	11.4	10.0	10.4	10.0	9.9	10.3	9.4	9.6	9.6	9.0	6.2	4.1	**4**	4.7	4.6	8.7
10	14.9	10.7	11.5	NC	11.0	13.8	10.6	11.0	10.4	10.2	9	6.8	**5.7**	6.7	7	9.8
15	31.9	11.7	12.1	11.7	11.2	31.2	11.5	11.6	11.1	11.3	20.7	8.9	**6.4**	8.1	7.8	10.5
20	77.4	11.6	11.8	11.1	10.9	74.6	10.9	11.6	10.5	10.8	24.8	9.9	**8.2**	9.3	10.3	12
Mean number of iterations to termination																
0	24.6	24.6	24.6	24.6	24.6	28.3	28.3	28.3	28.3	28.3	8.4	7.7	7.7	7.7	6.5	8
5	26.5	25.9	30.9	28.2	25.4	35.0	29.3	33.9	30.4	30.7	6.9	4.2	4.1	5	4.9	8.9
10	32.5	23.7	33.4	NC	26.9	44.1	27.9	35.1	31.3	27.9	7.5	4.8	4.5	5.6	6.6	9.4
15	50.3	27.0	48.8	34.4	25.0	59.7	30.5	47.5	37.4	29.1	8.8	6.3	4.9	5.4	6.4	9
20	44.3	25.4	54.6	33.9	26.6	42.8	29.3	55.8	38.2	29.7	9.4	7.4	5.5	6.6	6.8	9
Standard deviation of misclassifications																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.78	0
5	1.08	0.67	1.17	0.82	1.37	1.34	1.17	1.43	1.43	1.25	1.56	2.25	1.39	1.72	1.79	0.97
10	3.18	1.57	1.72	NC	1.70	2.66	1.78	1.89	1.65	1.75	5.24	3.28	0.78	1.75	2.11	1.64
15	31.72	3.06	2.77	3.27	2.35	30.76	3.14	2.68	2.92	2.21	8.63	1.85	1.82	2.14	2.16	1.39
20	57.11	3.13	3.46	3.28	2.73	48.75	2.56	3.06	2.87	2.78	9.73	3.16	2.71	2.27	2.09	2.69

improvement results range is between 14.84% and 32.77%. In addition, POK-means has worse results than ZF for low rates (0% to 5%) but better results by up 58.76% for higher rates (10%to 20%) for the Iris dataset. The results for the Wine dataset indicate a general improvement of the mean number of misclassifications that is up to 84.50% compared to kernel based clustering techniques. However, for this dataset POK-means does not come with an improvement of the misclassification results compared to imputation based clustering techniques. The best technique for this dataset is EMF. The results for the wholesale dataset show an improvement up to 23.89% and 41.20% for most of the missing rates compared to both kernel and imputation based clustering techniques respectively. IKCFM, KNNF,

Table 3 Accuracy and performance experimental results for incomplete Wholesale dataset

Rate	WDS	PDS	OCS	NPS	NNI	KWDS	KPDS	KOCS	KNPS	IKFCM	KMID	KNNF	EMF	MF	ZF	POK-means
Mean number of misclassifications																
0	61	61	61	61	61	54	54	54	54	54	59.7	51	51	51	64.7	**51**
5	63.3	62.0	60.7	61.3	59.0	52.4	53.0	52.7	53.1	**51.2**	52.2	52.3	53.2	66.4	51.8	52.6
10	66.4	62.8	60.5	61.4	57.9	51.8	53.2	52.5	53.2	**49.3**	69.1	51.8	50.7	68.6	66.5	54
15	72.0	63.7	59.9	60.9	59.5	55.7	54.3	53.9	54.2	**53.8**	57.1	54.7	51.5	70.5	93.2	54.8
20	72.1	67.8	63.4	65.8	64.0	60.5	56.4	56.0	56.5	61.1	70.4	57	**54.9**	58.4	83.7	56
Mean number of iterations to termination																
0	33.2	33.2	33.2	33.2	33.2	30.7	30.7	30.7	30.7	30.7	6.9	9	9	9	8	9
5	35.0	33.8	47.9	36.6	34.7	31.0	33.3	45.9	35.8	33.2	4.9	5.7	8.3	6.8	6.8	8.8
10	38.2	36.0	66.2	38.7	33.2	32.6	32.8	61.6	37.8	32.2	6.7	7.4	7.2	6.8	8	8.9
15	36.9	34.6	70.7	43.3	32.2	32.3	33.6	63.2	44.7	33.7	6	6.1	6.4	8.2	7.6	9.3
20	41.9	36.5	98.1	48.1	33.8	37.8	34.6	62.5	45.1	34.3	5.5	9.1	8.5	6.7	7.7	9.3
Standard deviation of misclassifications																
0	0	0	0	0	0	0	0	0	0	0	40.55	0	0	0	47.54	0
5	5.91	2.40	2.31	1.89	2.36	1.65	1.76	1.64	2.02	1.23	2.78	1.68	1.41	2.74	44.87	1.51
10	8.14	2.78	2.59	2.41	2.60	3.46	2.57	2.42	2.94	3.37	44.78	1.54	2.26	45.01	40.85	1.83
15	10.19	3.02	2.56	2.73	3.57	3.17	2.31	2.38	2.62	5.79	8.64	3.26	3.13	41.75	61.16	2.66
20	13.18	4.66	3.17	3.68	7.32	6.38	3.31	3.30	3.81	6.66	40.49	2.90	2.99	6.31	52.73	3.6

Table 4 Misclassification and performance experimental results for incomplete breast cancer dataset

Rate	WDS	PDS	OCS	NPS	NNI	KWDS	KPDS	KOCS	KNPS	IKFCM	KMID	KNNF	EMF	MF	ZF	POK-means
Mean number of misclassifications																
0	30	30	30	30	30	24	24	24	24	24	26.5	27	27	27	32.1	**23**
5	28.7	30.4	29.7	29.9	29.0	24.7	24.5	24.4	24.5	24.4	28.5	27.9	27.2	27.7	32.5	**22.8**
10	28.8	29.8	28.6	28.9	29.1	41.2	26.4	26.4	26.7	24.9	33.1	28.9	27.4	27.7	31.5	**23.4**
15	30.1	31.8	30.3	31.1	29.7	26.8	28.0	27.9	28.7	25.7	34.5	30.1	29	29.5	33.2	**24.7**
20	31.1	31.7	30.2	31.1	29.9	60.3	27.7	27.0	28.0	26.0	39	33.1	30.5	31.1	35.4	**25.4**
Mean number of iterations to termination																
0	12.6	12.6	12.6	12.6	12.6	26.4	26.4	26.4	26.4	26.4	4.4	3.6	3.6	3.6	5.1	6
5	12.6	12.5	29.8	13.8	12.9	24.9	26.2	29.9	26.7	26.4	5.2	4.1	4.1	4.2	5.8	6
10	12.8	12.9	39.1	15.3	13.3	28.1	26.8	36.7	29.5	26.5	6	4	4	4.4	5.3	6
15	12.8	13.1	47.9	16.4	13.0	26.1	27.4	38.1	30.8	26.3	5.8	4	4.3	3.9	4.9	6.2
20	13.0	13.5	43.3	18.8	12.8	30.5	28.9	62.6	36.0	24.2	6.4	3.9	3.9	4.6	6.5	5.3
Standard deviation of misclassifications																
0	0	0	0	0	0	0	0	0	0	0	0.52	0	0	0	1.66	0
5	2.91	1.65	1.70	1.97	1.63	2.06	1.78	1.71	1.78	1.78	4.19	2.51	1.47	2.54	4.47	2.1
10	2.66	1.62	2.46	2.08	2.77	49.53	2.46	2.41	2.54	1.73	6.08	2.13	2.59	1.25	4.06	1.96
15	1.97	2.82	3.50	3.07	2.71	2.39	1.94	2.08	1.34	2.36	8.64	2.18	2.05	2.27	4.10	2.75
20	2.77	3.06	3.08	3.45	3.54	67.54	2.11	2.05	2.87	2.67	9.72	3.24	3.27	2.33	3.47	2.99

EMF and POK-means are the best four techniques for this dataset. For the Breast Cancer dataset, the misclassification results show a clear improvement that is up to 43.20% and 29.85% compared to kernel and imputation based clustering techniques respectively. The improvement is obtained for all the missing rates for this dataset. The table shows that POK-means is the best technique for this dataset.

Regarding the stability of the clustering results, POK-means enjoys very low standard deviation values for the clustering results for the all datasets compared to both kernel and imputation based clustering techniques. This shows that the stability of POK-means is not affected by the increase of the data missing rate.

The results in terms of the mean number of iterations for termination indicate that POK-means has better performance than kernel based clustering techniques and is competitive to imputation based clustering techniques. This can be explained by the fact that the imputation allows the clustering algorithm to converge in fewer steps. However, this comes with a stability issue as aforementioned.

Figures 5, 6 and 7 outline the evaluation of POK-means clustering for the three datasets. The used metrics are: Accuracy, Normalized Mutual Information (NMI), F-score. The plots show, for the Iris and Breast Cancer datasets, that POK-means enjoys higher accuracy, NMI and F-score values than imputation based clustering techniques. More precisely, POK-means is among the three best techniques for the Iris, Wholesale and Breast Cancer datasets. However, POK-means is better only than KMID for the Wine dataset despite enjoying high accuracy, NMI and F-score values. An important feature of POK-means is that it enjoys better stability in terms of accuracy results compared to the other two superior techniques KNNF and EMF. Indeed, despite increasing the missing data rate, the values do not change drastically and the plots have almost a steady trend.

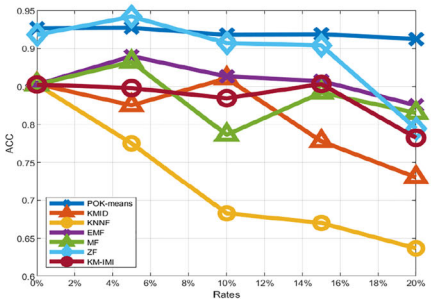
Figures 8 and 9 provide an overall view of comparison between POK-means, imputation [24] and the adaptive kernel based clustering algorithm [18] for the Iris and Thyroid Gland datasets. The figures show clearly that POK-means is competitive to these algorithms based on the accuracy and F-score metrics.

Based on the above results, we found that POK-means is ranked Best in 27 cases (out of 93) and Second Best in 5 cases. So, in almost 35% of the cases POk-means is either first or second. This shows that POK-means is competitive to other algorithms. Furthermore, we can see from the tables that the ranking of our algorithm generally improves when the missing ratio increases.

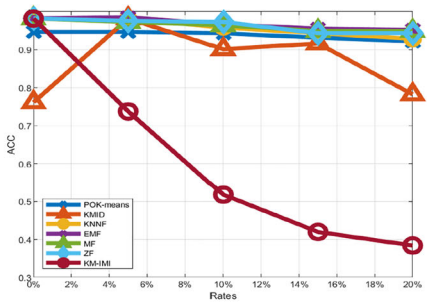
4.3 Scalability

To test the scalability of POK-means, we resorted to three large datasets: SYN, COMBO, and BENIGN TRAFFIC [15]. These datasets are related to the detection of IoT botnets or IoT traffic anomalies. COMBO includes 58,152 instances with 115 attributes. SYN has 118,129 instances with 115 attributes while BEGNIN TRAFFIC includes 175,240 instances with 115 attributes.

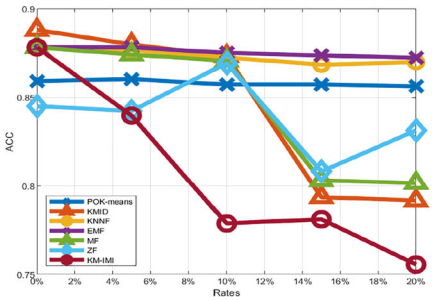
Fig. 5 Accuracy results



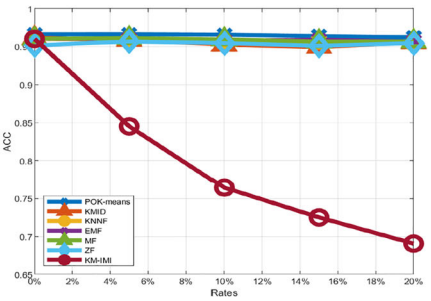
(a) Iris dataset



(b) Wine dataset

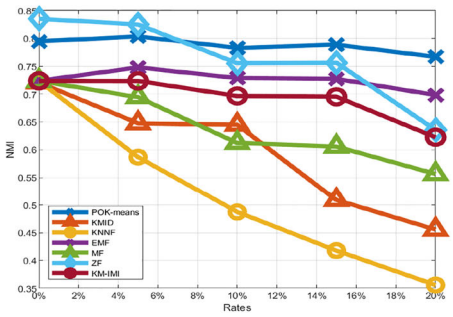


(c) Wholesale dataset

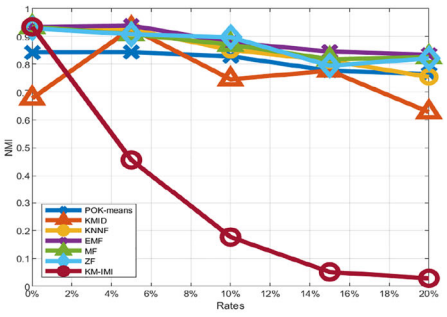


(d) Breast Cancer dataset

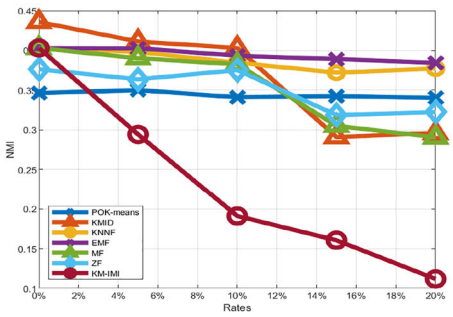
Fig. 6 NMI results



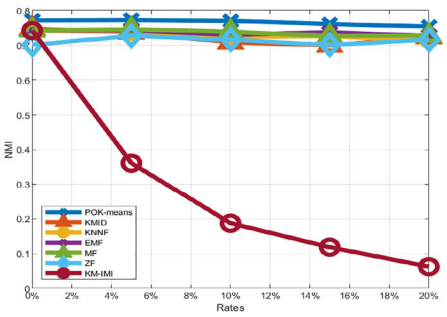
(a) Iris dataset



(b) Wine dataset

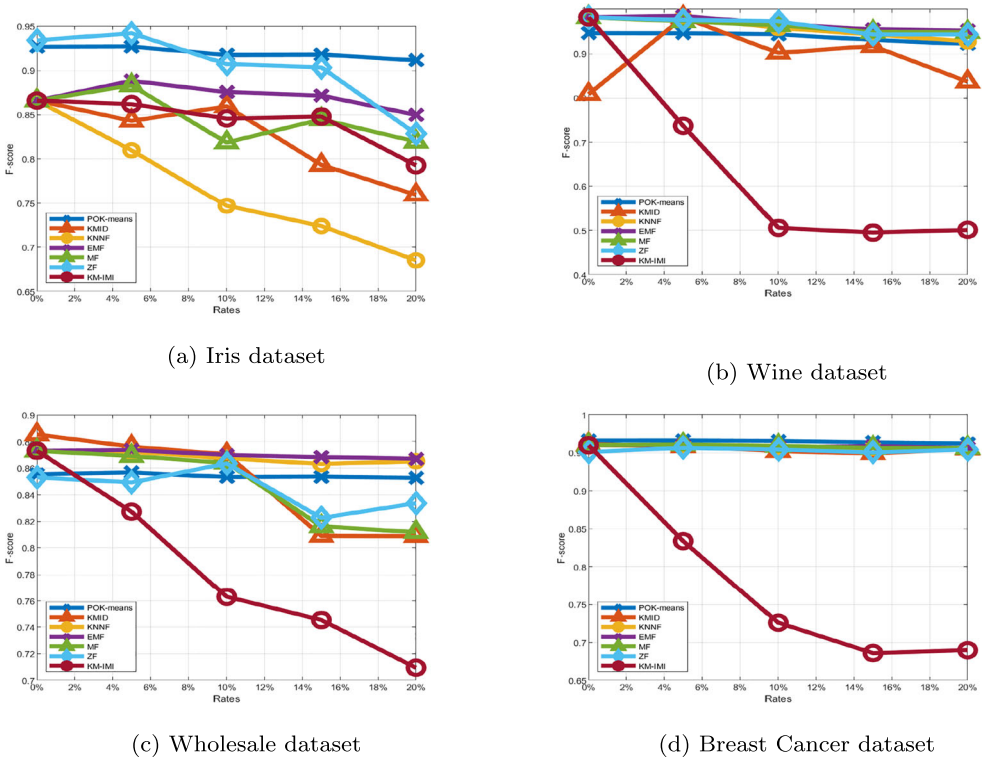


(c) Wholesale dataset



(d) Breast Cancer dataset

Fig. 7 F-score results



(a) Iris dataset

(b) Wine dataset

(c) Wholesale dataset

(d) Breast Cancer dataset

POK-means is compared with KNN-3, which is KNN with three neighbors and endowed with the Partial Distance to handle the missing values. The clustering time is measured for a certain partition of the data that varies between 20% and 100%.

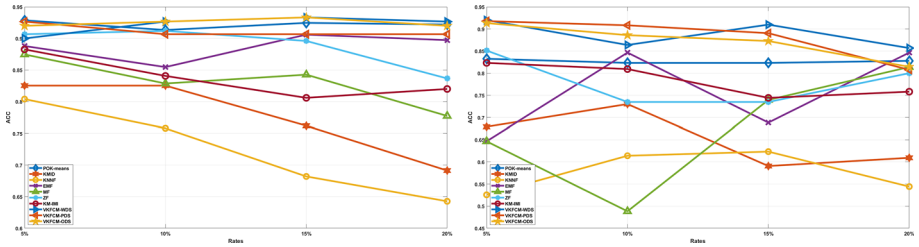
Figure 10 shows the scalability plots using a logarithmic scale for KNN-3 and POF. The results show that when the dataset partition increases in size, POF enjoys better scalability thanks to its fast convergence compared to KNN-3.

5 Conclusion

We proposed in this paper a new Partial Order-based Framework (POF) for clustering incomplete data. Our work was motivated by the shortcomings of existing methods such as [11, 12, 24], in particular, those related to the false dismissal problem. POF addresses this problem using the concept of interval distance and the underlying partial order. POF can guarantee the satisfaction of the lower bounding constraint for some distance approximations such as the weighted distance POWD. Furthermore, POF is a generic framework since it can handle complete and incomplete data, as shown in Section 3.4, and any clustering algorithm can be embedded in it.

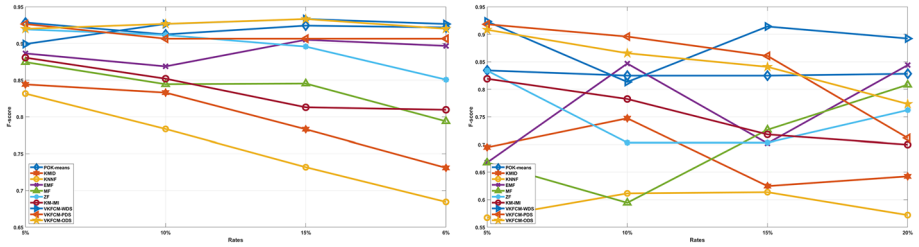
To illustrate the applicability of POF, we presented POK-means, which is an embedding of the K-means clustering algorithm in POF to deal with incomplete data. We further discussed its convergence. Our experimental results show that POK-means is competitive to recently published works.

Fig. 8 Accuracy results for POK-means, imputation and adaptive kernel clustering algorithms



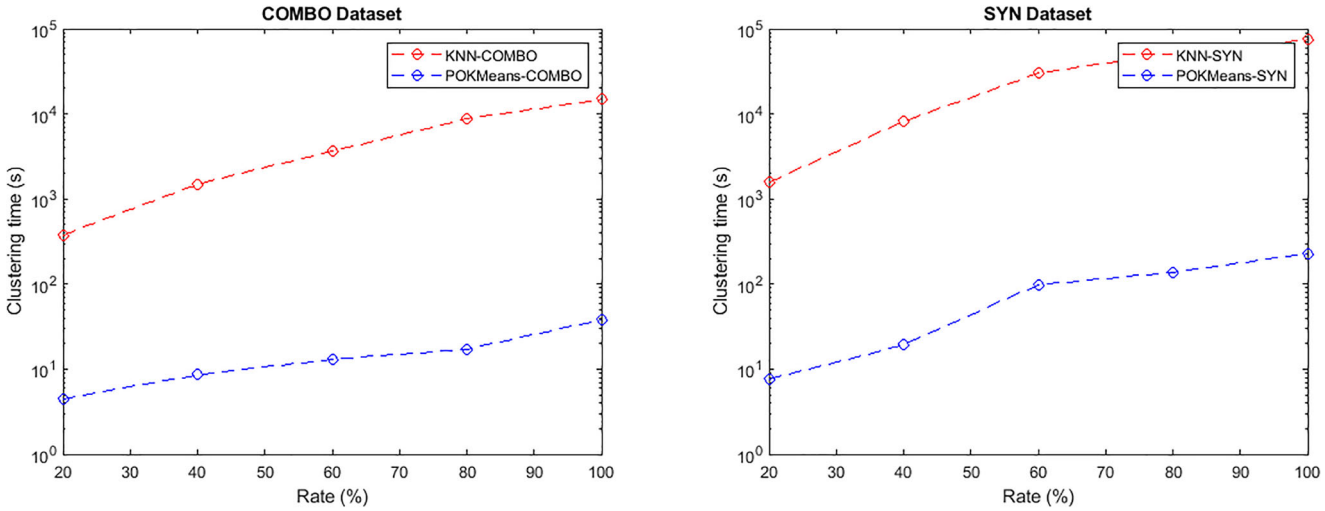
(a) Iris dataset

(b) Thyroid dataset

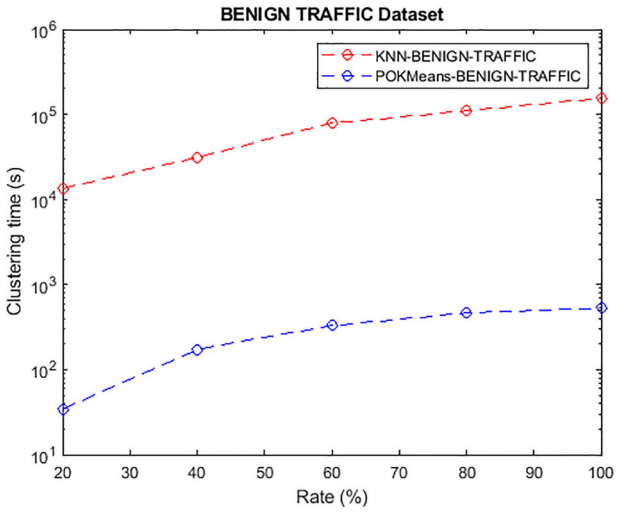


(a) Iris dataset (b) Thyroid dataset

Fig. 9 F-score results for POK-means, imputation and adaptive kernel clustering algorithms



(a) SYN dataset (b) COMBO dataset



(c) BENIGN dataset

Fig. 10 Scalability results for SYN, COMBO and BENIGN

One of the issues faced by POK-means is about antichains, which include incomparable elements in the lattice of interval distances. Indeed, POK-means could be stuck in one of antichain elements while applying the partial order on these intervals. This could lead to a bad clustering solution. We are planning to investigate this issue in our future work. An interesting research challenge consists in devising a good selection technique of antichain elements that can guarantee convergence to a good clustering solution. We are also planning to embed more clustering algorithms in the proposed framework.

Appendix A

Proposition 1 *$\preceq$ is a partial order.*

Proof –
- Reflexivity: $[a_L, a_R] \preceq [a_L, a_R]$ since $a_L \leq a_L \wedge a_R \leq a_R$.
- Asymmetry: Let us assume that $[a_L, a_R] \preceq [a'_L, a'_R]$ and $[a'_L, a'_R] \preceq [a_L, a_R]$. This means that $a_L \leq a'_L$ and $a'_L \leq a_L$. Hence, $a_L = a'_L$. By similar reasoning, we have $a_R = a'_R$. So $[a_L, a_R] = [a'_L, a'_R]$.
- Transitivity: Let us assume that $[a_L, a_R] \preceq [a'_L, a'_R]$ and $[a'_L, a'_R] \preceq [a''_L, a''_R]$. This means that $a_L \leq a'_L$ and $a'_L \leq a''_L$. So $a_L \leq a''_L$. By similar reasoning, we have $a_R \leq a''_R$. Therefore, $[a_L, a_R] \preceq [a''_L, a''_R]$.

□

Proposition 2 *$(\mathcal{L}_\mathcal{I}, \preceq)$ is a complete lattice.*

Proof Let $S \subseteq \mathcal{L}_\mathcal{I}$ such that $S = \{[a_1, b_1], [a_2, b_2], \ldots, [a_n, b_b]\}$. Then, an upper bound of the elements in S is $u = [\max(a_i), \max(b_i)]$. Let $[x, y]$ another upper bound of S. Based on the partial order $\preceq$, we can prove easily that $\max(a_i) \leq x$ and $\max(b_i) \leq y$, which means that $[\max(a_i), \max(b_i)] \preceq [x, y]$. Therefore, u is the least upper bound of S. By similar reasoning, S has a greatest lower bound $l = [\min(a_i), \min(b_i)]$. □

Theorem 1 *Any approximation based on the weighted distance POWD satisfies the lower bounding constraint.*

Proof Let d_i be an approximation value of a distance between two multidimensional data x and y for the i^{th} feature. We assume that d_i belongs to the interval distance $[d_{\min} = \sum_i d^i_{\min}, d_{\max} = \sum_i d^i_{\max}]$ in POF. This means that $d_i \leq d_{\max}$. So, $w_i\ d_i \leq w_i\ d_{\max}$. Thus, $\sum w_i\ d_i \leq (\sum w_i)\ d_{\max}$. Since by the definition of *POWD*, we have $\sum w_i = 1$, we conclude that *POWD* satisfies the lower bounding constraint. □

Theorem 2 *POK-Means in its strict version converges to a local minimum of the lattice $(\mathcal{L}_\mathcal{I}, \preceq)$.*

Proof The convergence proof is based on the following facts:

- The set of $(\mathcal{L}_\mathcal{I}, \preceq)$ is a complete lattice.
- In each iteration the SSE is minimized. So the sequence of iterations leads to a decreasing chain in term of SSE. Let S be the set of $I_0, I_1, \ldots, I_n$. S is finite since the number of configurations is finite. Since S is a subset of I has a greatest lower bound.

□

References

1. Basten T, Bosnacki D, Geilen M (2004) Cluster-based partial-order reduction. Autom Softw Eng 11:365–402
2. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood estimation from incomplete data via the em algorithm. J R Stat Soc Ser B 39:1–38
3. Dinh D, Huynh V, Sriboonchitta S (2021) Clustering mixed numerical and categorical data with missing values. Inf Sci 571:418–442
4. Dua D, Graff C (2019) UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences, http://archive.ics.uci.edu/ml. Last visit: May 2022
5. Fahad A, Alshatri N, Tari Z, Alamri A, Khalid I, Zomaya A, Foufou S, Bouras A (2014) A survey of clustering algorithms for big data: taxonomy and empirical analysis. IEEE Trans Emerg Top Comput 2:267–279
6. Faloutos C, Ranganathan M, Manolopoulos Y (1994) Fast subsequence matching in time-series databases. SIGMOD Rec 23:419–429
7. Hathaway R, Bezdek J (2001) Fuzzy c-means clustering of incomplete data. IEEE Trans Syst Man Cybern B 31:735–744
8. Hendriksen M, Francis A (2020) A partial order and cluster-similarity metric on rooted phylogenetic trees. J Math Biol 80:1265–1290
9. Kang H (2013) The prevention and handling of the missing data. Korean J Anesthesiol 64:402–406
10. Kline RB (2015) Principles and Practices of Structural Equation Modeling, Fourth Edition. Guilford Press, New York. ISBN: 978-1-4625-2335-1
11. Li D, Gu H, Zhang L (2010) A fuzzy c-means clustering algorithm based on nearest-neighbor intervals for incomplete data. Expert Syst Appl 37:6942–6947
12. Li T, Zhang L, Wei L, Hou H, Liu X, Pedrycz W (2017) Interval kernel fuzzy c-means clustering of incomplete data. Neurocomputing 237:316–331
13. Lin J, Keogh E, Wei L, Leonardi S (2007) Experiencing SAX: a novel symbolic representation of time series, vol 15
14. Matyja A, Siminski K (2014) Comparison of algorithms for clustering incomplete data. Found Comput Decis Sci 39:107–127

15. Meidan Y, Bohadana M, Mathov Y, Mirsky Y, Breitenbacher D, Shabtai A, Elovici Y (2018) N-baiot: network-based detection of iot botnet attacks using deep autoencoders. IEEE Pervasive Computing, Special Issue - Securing the IoT 17:12–22
16. Quilan JR (1986) Induction of decision trees. Mach Learn 1:81–106
17. Raskin A (2014) Comparison of partial orders clustering techniques. Proc ISP RAS 26:91–98
18. Rodrigues A, Ospina R, Ferreira M (2021) Adaptive kernel fuzzy clustering for missing data. PLoS ONE 16:1–33
19. Rodriguez M, Comin C, Casanova D, Bruno M, Amancio D, Costa L, Rodrigues A (2019) Clustering algorithms: a comparitive approach. PLoS ONE 14:1–34
20. Sammut C, Webb G (2017) Encyclopedia of Machine Learning and Data Mining, Second Edition. Springer, New York. ISBN: 978-1-4899-7685-7
21. Schafer JL, Olsen MK (1998) Multiple imputation for multivariate missing data problems: a data analyst's perspective. Multivar Behav Res 33:545–571
22. Schlomer G, Bauman S, Card N (2010) Best practices for missing data managemnt in counseling psychology. Journal of Counseling Psychology American Psychological Assocition 57:1–10
23. Shi H, Wang P, Yang X, Yu H (2020) An improved mean imputation clustering algorithm for incomplete data. Neural Process Lett, https://doi.org/10.1007/s11063-020-10298-5
24. Siwei W, Miaomiao L, Ning H, En Z, Jingtao H, Xinwang L, Jianping Y (2019) K-means clustering with incomplete data. IEEE Access 7:69162–69171
25. Tellaroli P, Bazzi M, Donato M, Brazzale AR, Drăghici S (2016) Cross-clustering: a partial clustering algorithm with automatic estimation of the number of clusters. PLoS ONE 11:1–14
26. Ukkonen A (2011) Clustering algorithms for chains. J Mach Learn Res 12:1389–1423
27. Zhang Y, Li M, Wang S, Dai S, Luo L, Zgu E, Xu H, Zhu X, Yao C, Zhou H (2021) K-Means Clustering with incomplete data. ACM Trans Multimed Comput Commun Appl 17:1–14