FOUNDATIONS OF COMPUTING AND DECISION SCIENCES
Vol. 39 (2014) No. 2

DOI: 10.2478/fcds-2014-0007 ISSN 0867-6356
e-ISSN 2300-3405

COMPARISON OF ALGORITHMS FOR CLUSTERING INCOMPLETE DATA

Artur MATYJA, Krzysztof SIMIŃSKI *

Abstract. The missing values are not uncommon in real data sets. The algorithms and methods used for the data analysis of complete data sets cannot always be applied to missing value data. In order to use the existing methods for complete data, the missing value data sets are preprocessed. The other solution to this problem is creation of new algorithms dedicated to missing value data sets.

The objective of our research is to compare the preprocessing techniques and specialised algorithms and to find their most advantageous usage.

Keywords: clustering, incomplete data, missing value, marginalisation, imputation, IFCM, OCS, NPS, NCS

1 Introduction

The missing values are not uncommon in real data sets. They occur due to the measurement errors or loss of values after acquisition. Many medical data lack some values [18]. Some data may be merged from various sources and they are not fully compatible or they are used with different aim than they were collected with.

The values may *miss* from the data set *completely at random* (MCAR) – the probability of an tuple having a missing value for an attribute depends neither on the known values nor on the missing data. In the second type of incompleteness the probability that the tuple has a missing value may depend on the known values, but not on the value of the missing data itself. The is *missing at random* (MAR) type of incompleteness. In the third type the values do *not miss at random* (NMAR) – the probability of an instance having missing value for an attribute can depend on the value of that attribute [15].

*Institute of Informatics, Silesian University of Technology, ul. Akademicka 16, 44-100 Gliwice, Poland, Krzysztof.Siminski@polsl.pl

Table 1: Symbols used in the papers. See page 2 for general rule for symbols.

$\mathbb{X}$	set of tuples, data examples, $\mathbb{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_X\}$
$\mathbf{x}$	vector of tuple's descriptors, data example, $\mathbf{x} \in \mathbb{X}$
X	number of tuples, $X = \|\mathbb{X}\|$
x	descriptor of a tuple, $\mathbf{x} = [x_1, \ldots, x_D]^{\mathrm{T}}$
k	object
K	number of objects
$\mathbb{K}$	set of objects
$\mathbb{D}$	set of attributes
d	attribute, $d \in \mathbb{D}$
D	number of attributes in a tuple, $D = \|\mathbb{D}\|$
$\mathbb{C}$	set of clusters
C	number of clusters, $C = \|\mathbb{C}\|$
c	cluster, $c \in \mathbb{C}$
$\mathbf{U}$	partition matrix, $\mathbf{U} = \{u_{ij}\}$
u_{ij}	membership value of the j-th tuple to i-th cluster
t_{ij}	distance between i-th cluster's centre and j-th tuple

The algorithms and methods used for the data analysis of complete data sets cannot always be applied to missing value data. In order to use the existing methods for complete data, the missing value data sets are preprocessed. The other solution to this problem is the creation of new algorithms dedicated to missing value data sets.

The objective of our paper is to compare various techniques on the same data sets and to determine which approach can discover the localisation of the data clusters in spite of the missing values in the data sets. As the reference cluster localisation we take the clusters elaborated from the complete data set.

In the paper we propose the dissimilarity measure for two cluster sets. We define it as a dissimilarity of localisation of the cluster centres. We plan to use the clustering as the step in creation of the fuzzy rule base for the neuro-fuzzy system. In this approach the localisation of the cluster centre is crucial, the fuzzification of the clusters is to be tuned in tuning procedure.

In the paper we discuss the algorithms for data sets with numerical attributes. The algorithms that take into account the missing attributes have been proposed for data sets with categorical attributes [10].

The paper is organised as follows: Section 2 shortly summarises the preprocessing techniques, Sec. 3 describes the specialised algorithm compared in our paper. Section 4 presents quality measures used in validation of the elaborated clusters. Section 5 describes the data sets and experiments and finally Sec. 6 summaries the paper.

In the paper we follow the general rule for symbols: the blackboard bold uppercase characters ($\mathbb{A}$) are used to denote the sets, uppercase italics (A) – the cardinality of sets, uppercase bolds ($\mathbf{A}$) – matrices, lowercase bolds ($\mathbf{a}$) – vectors, lowercase italics (a) – scalars and set elements. Table 1 lists the symbols used in the paper.

2 Data preprocessing

The data preprocessing is a common technique for handling incomplete data with methods that proved to be efficient with complete data [9]. This step is not always necessary. In the paper we describe the specialised algorithms for incomplete data, many others techniques have been developed [10]. There are two essential preprocessing techniques: missing values are marginalised or imputed with some values.

Marginalisation (WDS – whole data strategy) removes either incomplete data vectors or the features with missing values. Deletion of the incomplete data vectors is more common than marginalisation of the features.

The missing values are imputed with various techniques: by simple imputation (with zeros, means, medians, random values) or more sophisticated approximation (regression [4, 23], expectation-maximization [6, 8], nearest neighbours [26, 27]).

Mean imputation is one of the most often applied methods. The missing value of an attribute is substituted with a mean of all existing values of this attribute in the whole data set. This method may impute non-existing values into the data vectors. The imputed values may have no physical meaning [23]. Mean imputation is vulnerable to outliers. Median imputation avoids these drawbacks: in this method missing values are imputed with existing values. The outliers have lower influence on median imputation. The disadvantage of this approach is longer time of median calculation in comparison to mean imputation.

The simple mean and median imputation techniques use the whole data set to calculate the mean or median values. For some data sets it is more advantageous to use a certain subset of tuples to calculate mean or median values. Mean or median is calculated basing only on k nearest neighbours. The distance function between two objects has to be defined. This method has two main drawbacks: the difficulty in choosing of optimal number (k) of neighbours and the high cost of finding neighbours.

Both marginalisation and imputation are commonly used due to their simplicity. Imputation is applied more frequently than marginalisation [14]. Preprocessed data become less reliable: marginalisation loses some information, imputation may add non-existing or meaningless information [23]. The imputed values are not labelled and indiscernible form original values what can also be treated as some loss in information. Biological conclusions from imputed data cannot be drawn with high reliability [22].

3 Specialised clustering algorithms

The second major approach to clustering of incomplete data is applying specialised and dedicated clustering algorithms. These algorithms use various techniques. Most of them incorporate imputation into clustering procedure.

The algorithm proposed in [20, 19] applies rough sets to clustering of data with missing values. This algorithm will not be further analysed in our paper as it elaborates the rough clusters.

Many specialised algorithms are based on the fuzzy c-mean (FCM) algorithm [7].

FCM clustering minimises the objective function

$$J(\mathbf{U},\mathbf{V}) = \sum_{c=1}^{C}\sum_{k=1}^{K}(u_{ck})^m \|\mathbf{x}_k - \mathbf{v}_c\|^2 \tag{1}$$

with constraints

$$\forall k \in \mathbb{K} : \sum_{c=1}^{C} u_{ck} = 1, \tag{2}$$

where C stands for number of clusters, K – number of objects (vectors), $\mathbf{x}$ – object (data vector), $\mathbf{v}$ – cluster centre and u_{ck} is membership values of the kth object to the cth cluster and $m > 0$ is weighting exponent [7]. There is no theoretical basis for fixing the m value, we use here $m = 2$ [5]. The centre of the cth cluster is elaborated with the formula

$$\forall c \in [1, C] : \mathbf{v}_c = \frac{\sum_{k=1}^{K}(u_{ck})^m \mathbf{x}_k}{\sum_{k=1}^{K}(u_{ck})^m} \tag{3}$$

The improved fuzzy c-means (IFCM) [21] imputes the missing values iteratively. The clusters are elaborated with full data examples, then the missing values are imputed with weighted mean of values of missing attributes from elaborated clusters' centres. The weights are the membership values of the object in question to the found clusters.

The Partial Distance Strategy (PDS) calculates the distance of the object to the cluster basing only on the existing values. The total distance is modified by the number of used dimensions. The distance t_{ck} of the kth object to cth cluster is defined as

$$t_{ck} = \sqrt{\frac{D\sum_{d=1}^{D}(x_{kd} - v_{cd})^2 I_{kd}}{\sum_{d=1}^{D} I_{kd}}}, \tag{4}$$

where

$$I_{kd} = \begin{cases} 1, & \text{if } d\text{th attribute in } k\text{th object exists,} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

and D stands for the number of attributes in the data vector. When the object lacks no values the distance measure defined above is Euclidean measure. This method modifies the formula for cluster centres (cf. Eq. 3):

$$\forall c \in [1, C] : \forall d \in [1, D] : v_{cd} = \frac{\sum_{k=1}^{K}(u_{ck})^m x_{kd} I_{kd}}{\sum_{k=1}^{K}(u_{ck})^m I_{kd}}. \tag{6}$$

The Optimal Completion Stategy (OCS) treats the missing values as additional variables. The algorithm is similar to FCM. First the cluster centres $\mathbf{v}_1, \ldots, \mathbf{v}_V$ and missing values in data set are initialised with random numbers. Then the new membership matrix $\mathbf{U}$ is calculated and new cluster centres. The missing values are recalculated with formula:

$$x_{kd} = \frac{\sum_{c=1}^{C}(u_{ck})^m v_{cd}}{\sum_{c=1}^{C}(u_{ck})^m}. \tag{7}$$

The Nearest Prototype Stategy (NPS) [13] is similar to the OCS. The missing values in an object k are recalculated, but instead of applying formula 7 the nearest prototype (object with all attributes) is found and the missing values in object k are substituted with respective values of the nearest prototype. For calculating distance the formula 4 is used.

The Nearest Cluster Strategy (NCS) [25] uses the Gustafson-Kessel [12] clustering. First all data with missing values are removed (marginalisation), the cluster centres are calculated. Then for each data vector $\mathbf{x}_i$ with missing values the nearest cluster centre $\mathbf{v}_n$ is found. When distance is calculated only the existing dimensions are used. Having discovered the nearest cluster n, the search for nearby vectors to this cluster is started. These vectors are closer to the discovered cluster n than to other clusters' centres. The missing values of the vector $\mathbf{x}_i$ are imputed with the mean values of the respective attributes of the vectors neaby to the discovered cluster n. Finally the whole set of vectors (original with all attributes and vectors with imputed values) are clustered. In our experiment we use version with the FCM clustering [7], so that the comparison of the algorithms is more reliable. We do not want to compare the FCM and the Gustafson-Kessel algorithms, but to compare the modifications of clustering algorithms for incomplete data.

4 Quality of clustering

In this section we shortly describe the indices for evaluating the quality of clustering (Sec. 4.1) and the proposition of an index for comparison of clusters elaborated by the two algorithms (Sec. 4.2).

4.1 Quality indices

There seems not to be one good index for evaluating clustering. Many indices have been proposed, in this section we shortly describe the measures as in [5]. Lower values of the indices denote higher quality of clustering. The simplest are *partition coefficient*

$$V_{\mathrm{PC}}(\mathbf{U}) = -\frac{1}{K}\sum_{c=1}^{C}\sum_{k=1}^{K}\left(u_{ck}\right)^2 \tag{8}$$

and *partition entropy*

$$V_{\mathrm{PE}}(\mathbf{U}) = -\frac{1}{K}\sum_{c=1}^{C}\sum_{k=1}^{K} u_{ck}\ln u_{ck}. \tag{9}$$

Some more sophisticated indices have also been proposed. Xie and Beni [24] proposed index defined as

$$V_{\mathrm{XB}}^{(m)}(\mathbf{U}) = \frac{\sum_{c=1}^{C}\sum_{k=1}^{K}\left(u_{ck}\right)^m t_{ck}^2}{K\left(\min_{i\neq j}\|\mathbf{v}_i-\mathbf{v}_j\|^2\right)} = \frac{J_m\left(\mathbf{U},\mathbf{V}\right)}{K\left(\min_{i\neq j}\|\mathbf{v}_i-\mathbf{v}_j\|^2\right)}. \tag{10}$$

Xie and Beni use $m = 2$. The index measures both compactness (nominator) and separation (denominator) of clusters [5].

Fukuyama and Sugeno proposed [16] index defined as

$$V_{\mathrm{FS}}^{(m)}(\mathbf{U}) = J_m(\mathbf{U}, \mathbf{V}) - \sum_{c=1}^{C} \left\{ \left[\sum_{k=1}^{K} (u_{ck})^m \right] \|\mathbf{v}_c - \mathbf{v}\|^2 \right\}, \tag{11}$$

where $\mathbf{v}$ stands for grand mean over all data vectors.

Bensaid proposed [3] following index

$$V_{\mathrm{B}}^{(m)}(\mathbf{U}) = \sum_{c=1}^{C} \frac{\sum_{k=1}^{K} (u_{ck})^m \|\mathbf{x}_k - \mathbf{v}_c\|^2}{n_c \sum_{j=1}^{C} \|\mathbf{v}_c - \mathbf{v}_j\|^2}. \tag{12}$$

Czogała and Łęski defined clustering validity index as mean quotient of dissipation against the cluster centre by dissipation of cluster centre against the centre of the given cluster [5]:

$$V_{\mathrm{CzŁ}}^{(m)} = \sum_{j=1}^{C} \frac{n_c \sum_{k=1}^{K} (u_{ck})^m \|\mathbf{x}_k - \mathbf{v}_c\|^2}{\sum_{k=1}^{K} (u_{ck})^m \sum_{j=1}^{C} (n_c + n_j) \|\mathbf{v}_j - \mathbf{v}_c\|^2}. \tag{13}$$

For all indices above lower values mean better clustering.

4.2 Comparison of clustering results

All indices presented above are designed for evaluating results of clustering procedure elaborated for one data set. We propose the cluster dissimilarity measure to compare the localisation of the clusters elaborated by various algorithms. We use this index to evaluate the clusters elaborated for incomplete data set in comparison with the clusters for the complete version of the data set. The pseudocode for cluster dissimilarity measure is presented in Fig. 1. The reference set of clusters is the set elaborated with FCM algorithm for complete data. The second set of clusters is calculated for incomplete data with one of the described above techniques. Then we sort the clusters of each set with the first attribute being the key of sorting. Next we calculate the Euclidean distances between the first clusters in each cluster sequences, then the distance between second pair etc. The distances are added up. We repeat the procedure for each dimension and select minimal value.

The number of all matches of two sets holding C clusters each is $C!$ (the exhaustive search requires an analysis of all permutations of set with C items). This is why we proposed the heuristic algorithm presented in Fig. 1. Its complexity is $O(DC(\log C + D)$, where D stands for number of dimensions, if we use $O(n \log n)$ sorting algorithm. However it should be taken into consideration that when $C \ll D$ it is more convenient to use the exhaustive search instead of heuristics. The Fig. 2 presents the example of calculation of the dissimilarity measure for two sets of three clusters.

```
1  function dissimilarity_measure (V1, V2);
2  input: V1, V2
3  {Matrices contain the centres of the clusters from two
       clustering procedures, an element v_cd is the localisation
       of cth cluster in dth dimension. The numbers of attributes
        in both cluster sets are the same (all_dimensions).}
4  output: dissimilarity {dissimilarity measure}
5  begin
6     dissimilarity := ∞;
7     for d := 1 to D do {for all dimensions}
8        {sort centre matrices by dimension d}
9        sort (V1, d);
10       sort (V2, d);
11       distance := 0; {sum of distances between all pairs of
             clusters}
12       for c := 1 to C do {for all clusters}
13          difference := 0; {between cluster centres in p-th
                dimension}
14          for p := 1 to D do {for each dimension}
15             difference += power(V1[c, p] - V2[c, p], 2);
16          end for;
17          distance += sqrt(difference);
18       end for;
19       if distance < dissimilarity then
20          dissimilarity := distance;
21       end if;
22    end for;
23    return dissimilarity;
24 end.
```

Figure 1: Algorithm for calculation of dissimilarity index.

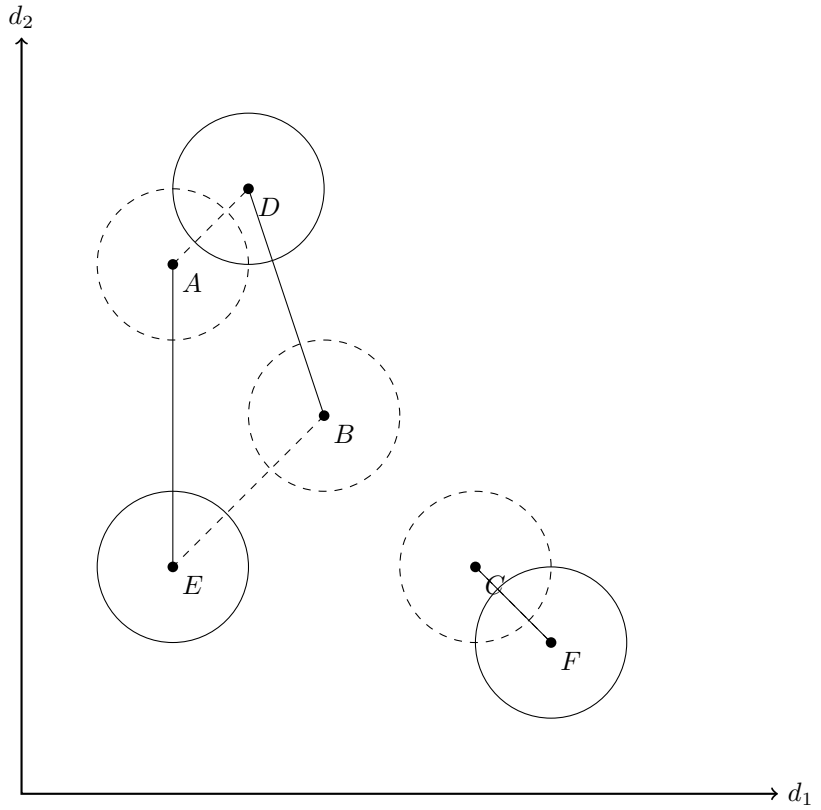

Figure 2: The figure illustrates the application of the dissimilarity measure described by the pseudocode in Fig. 1. Let's assume we have two sets of clusters in two dimensional space: $\mathbb{C}_1 = \{A, B, C\}$ (dashed circles) and $\mathbb{C}_2 = \{D, E, F\}$ (solid circles). The algorithm tries to match the clusters in both sets in all dimensions. First the clusters are match in the d_1 dimension, the clusters in both sets are sorted and we get two series: (A, B, C) for the $\mathbb{C}_1$ and (E, D, F) for the $\mathbb{C}_2$. The matches $\{A, E\}$, $\{B, D\}$ and $\{C, F\}$ are denoted by solid lines. The distances in this match are summed up. Then the procedure is repeated for the next dimension d_2. In this case the clusters are sorted into series (C, B, A) for the $\mathbb{C}_1$ and (F, E, D) for the $\mathbb{C}_2$. The matches are $\{C, F\}$, $\{B, E\}$ and $\{A, D\}$. The distances (denoted by dashed lines, for $\{C, F\}$ the dashed line is overprinted by the solid one from the first match) are summed up. The minimal value of sums is returned as the dissimilarity measure.

5 Experiments

The experiments were conducted on real life data sets.

We used preprocessing techniques described above with subsequent FCM and specialised algorithms for clustering incomplete data. The reference cluster set was elaborated with FCM and complete data. The complete data set contains the whole information and incomplete data sets miss some information. This approach seems reasonable although it has been shown in [11] that rules induced from incomplete data sets can be more useful than rules induced from complete data sets. It requires more research to check whether similar situation may occur in clustering task.

The stop condition of algorithms is an alternative of two subconditions:

- $\max_{i,k} \left\| u_{ik}^{(r)} - u_{ik}^{(r-1)} \right\| \leqslant \varepsilon$, where r stands for the number of iterations or
- $r > R$, where R is the maximal number of iterations.

Our objective was to test whether various algorithms can discover similar localisation of clusters or whether each of them discovers totally different clusters for the same data sets. To check this we tried to find the best match of clusters in two cluster sets. We used the algorithm described in Sec. 4.2.

5.1 Data sets

The 'Iris' data set is commonly known and consists of 150 samples of Iris plants divided into three classes. Each data tuple is represented by 4 attributes (sepal length, sepal width, petal length and petal width).

The 'Glass' data set [2] is an imbalanced version of the Glass data set, where the positive examples belong to class 2 and the negative examples belong to the rest. The data set can be downloaded from public repository[1]. The data set has 214 instances of 9-feature tuples.

The 'Telugu' data set [17] describes Telugu vowels. They are characterised by 3 attributes (frequency values) and class ('a', 'e', 'i', 'u', 'o' and 'ə' vowels). The data set contains 871 vowels.

The incomplete data sets were created from the data described above. The ratio of missing value is: 1, 2, 3, 5, 7, 10, 15, 25%. The value miss at random in three patterns:

- form the whole data set (NMAR),
- only from one attribute, other attributes are complete,
- only from one cluster, other clusters are complete (MAR).

Each approach was started 10 times for each option (missing value ratio and the missing pattern), the averages of indices, times and number of iterations are presented in tables and figures.

[1] `http://sci2s.ugr.es/keel/dataset.php?cod=121`

5.2 Results

The experiments were conducted for following values of the ε parameter: 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}. For the brevity of the paper we will limit the presented results only to $\varepsilon = 10^{-4}$. One value shows the results sufficiently good.

In the experiments we use applied preprocessing ('mar' – marginalisation of objects, '$\bar{x}$' – average imputation, '$\tilde{x}$' – median imputation, '$k\bar{x}$' – k nearest neighbours average imputation, '$k\tilde{x}$' – k nearest neighbours median imputation) and subsequent FCM algorithms and specialised algorithm: IFCM, PDS, OCS, NPS and NCS.

The results of the experiments on the 'Iris' data set are presented in Tab. 2. The table presents the dissimilarity measure (defined in Sec. 4.2), clustering time (in milliseconds) and the number of iterations. The results are also visualised in Figures 3 (dissimilarity), 4 (clustering time) and 6 (number of iterations). We provide the results for the time of clustering and the number of iterations only for the 'Iris' data set. The results for the 'Glass' and 'Telugu' data sets are similar and we restrain ourselves from pasting very similar results. The Table 3 presents the dissimilarity index for the 'Glass' data set. The tables 12, 16, 14, 17, 15 and 13 present the quality measures for the 'Telugu' data set.

The results show that the for small ratios of missing values the best results are elaborated by median imputation (both simple and k nearest neighbours). The specialised clustering algorithms take about twice more time than preprocessing and FCM algorithm. The results show that the number of iteration is lower in preprocessing and complete data clustering than in the specialised algorithms.

For higher ratios of missing values ($> 10\%$) the specialised algorithms are more advantageous. The NCS algorithm has the longest time of clustering. It is worth mentioning that even for high missing value ratio (25%) marginalisation is better than all imputation method we have analysed, but worse than specialised algorithms. Marginalisation is the quickest method for high missing ratios, this seems natural. The analysed methods are independent of the kind of missing pattern (described in Sec. 5.1).

The cluster quality indices are not unanimous in case of the specialised algorithms. The Fukuma-Sugeno index points the IFCM algorithm as the best one, whereas the Bensaid index points this algorithm as the worst one. For the 'Iris' data set the Xie-Beni and Czogała-Łęski indices do not show unambiguously the most advantageous algorithm for the specialised ones. For the 'Telugu' data set the Czogała-Łęski index orders the algorithm from the best OCS through IFCM to the worst PDS algorithm. The dissimilarity index is in concordance with the Czogała-Łęski, Bensaid and Xie-Beni indices. Basing on the clustering indices it cannot be stated which specialised algorithm elaborates the best results. These algorithms differ severely in times of calculation. The IFCM differs by one and the NCS by two orders of magnitude in time of calculation from the PDS, OCS and NPS algorithms. The dissimilarity measures is in concordance with partition entropy (Eq. 9), Xie-Beni (Eq. 10), Fukuyama-Sugeno (Eq. 11) and Czogała-Łęski (Eq. 13) indices.

The results achieved for the 'Iris' dataset with missing values from one cluster are gathered in Tab. 4. The experiments were repeated for all data sets with missing

Table 2: The dissimilarity measure between clusters elaborated for the complete and incomplete 'Iris' data set. The results are also drawn in figures 3 (dissimilarity measure), 6 (number of iterations), 4 and 5 (time of clustering). The left part of the table shows the results for preprocessed data and subsequent FCM algorithm. The preprocessing techniques are labelled: 'mar' – marginalisation of objects, '$\bar{x}$' – average imputation, '$\tilde{x}$' – median imputation, '$k\bar{x}$' – k nearest neighbours average imputation, '$k\tilde{x}$' – k nearest neighbours median imputation.

	FCM					specialised algorithms				
%	mar	$\bar{x}$	$\tilde{x}$	$k\bar{x}$	$k\tilde{x}$	IFCM	PDS	OCS	NPS	NCS
					dissimilarity measure					
1	0.31	0.16	0.02	0.16	0.02	0.31	0.13	0.11	0.11	0.11
2	0.17	0.04	0.04	0.03	0.04	0.03	0.13	0.11	0.12	0.11
3	0.17	0.07	0.06	0.18	0.08	0.31	0.13	0.12	0.13	0.12
5	0.47	0.13	0.10	0.07	0.13	0.18	0.13	0.13	0.14	0.13
7	0.20	0.16	0.14	0.10	0.50	0.23	0.13	0.13	0.14	0.12
10	0.00	0.25	0.22	0.17	0.41	0.07	0.14	0.16	0.15	0.14
15	0.50	0.38	0.38	0.29	0.76	0.12	0.13	0.17	0.17	0.14
25	0.36	0.84	1.05	0.58	1.13	0.36	0.15	0.24	0.22	0.28
					clustering time [ms]					
1	2.61	3.47	2.98	3.36	2.49	55.87	7.47	5.70	7.74	141.67
2	2.39	2.94	3.05	3.30	2.78	68.64	8.29	6.70	6.97	255.98
3	2.63	2.92	3.04	3.49	2.95	64.79	8.25	6.57	5.91	348.65
5	2.28	3.43	3.06	3.48	2.95	75.59	7.26	7.14	8.22	421.01
7	2.07	3.23	3.10	4.00	4.73	75.13	8.06	7.37	9.13	570.38
10	2.21	3.48	2.94	3.25	3.74	100.17	7.99	8.37	7.02	815.09
15	1.33	3.31	3.36	3.31	4.81	113.91	9.59	9.81	8.54	985.08
25	1.27	4.96	5.67	4.65	4.21	118.58	7.49	14.55	18.36	1081.23
					number of iterations					
1	16.20	18.50	12.90	13.80	14.60	182.00	20.60	15.70	14.70	17.80
2	14.20	15.40	14.10	14.30	14.20	200.00	21.60	15.80	16.00	15.60
3	15.30	15.60	14.10	15.00	17.30	200.00	21.50	17.80	35.90	16.80
5	17.20	14.80	12.40	14.10	19.10	200.00	20.90	19.90	18.00	15.20
7	14.70	13.70	13.60	14.00	22.70	200.00	23.20	24.00	35.40	16.00
10	14.10	14.90	12.80	13.10	25.70	200.00	20.90	26.50	21.30	15.80
15	18.60	15.40	15.50	13.80	21.60	200.00	23.70	25.30	22.70	18.40
25	14.90	26.90	47.60	22.80	17.30	200.00	21.60	37.30	26.10	17.20

Table 3: The dissimilarity index between cluster sets for the complete 'Glass' data set and incomplete sets with various missing value ratio. The abbreviations are the same as in Tab. 2.

	FCM					specialised algorithms				
%	mar	$\bar{x}$	$\tilde{x}$	$k\bar{x}$	$k\tilde{x}$	IFCM	PDS	OCS	NPS	NCS
1	0.007	0.062	0.062	0.054	0.056	0.017	0.182	0.248	0.248	0.248
2	0.016	0.189	0.191	0.192	0.192	0.012	0.011	0.251	0.250	0.250
5	0.024	0.234	0.237	0.235	0.238	0.012	0.009	0.243	0.240	0.236
10	0.037	0.297	0.307	0.301	0.305	0.077	0.012	0.280	0.248	0.254
15	0.098	0.335	0.338	0.335	0.339	0.277	0.225	0.245	0.246	0.248
25	0.091	0.334	0.341	0.336	0.340	0.230	0.018	0.288	0.273	0.225

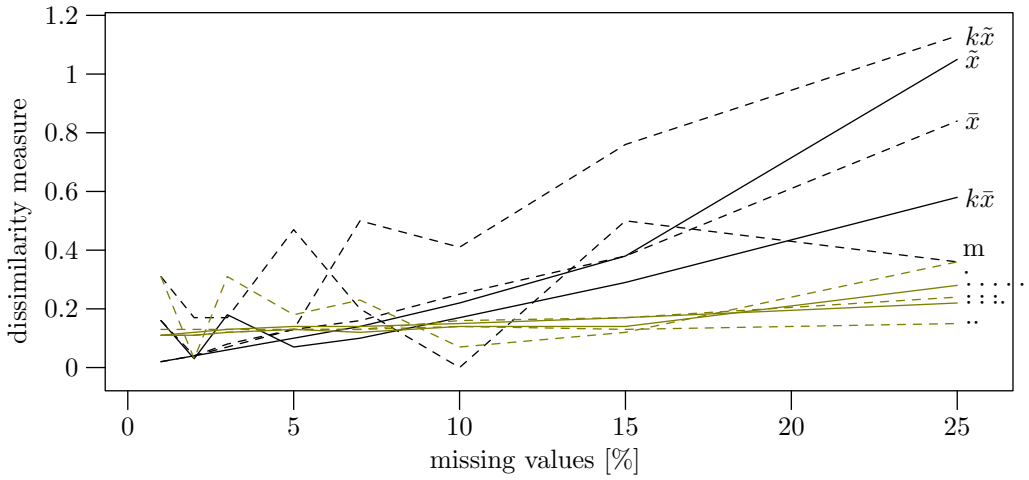

Figure 3: The dissimilarity measure between clusters elaborated for the complete and incomplete 'Iris' data set. The results are gathered in Tab. 2. Abbreviations: 'm' – marginalisation of objects, '$\bar{x}$' – average imputation, '$\tilde{x}$' – median imputation, '$k\bar{x}$' – kNN average imputation, '$k\tilde{x}$' – kNN median imputation. The specialised algorithms are denoted with dots (·) in following convention: (·) IFCM, (··) PDS, (· · ·) OCS, (· · ··) NPS, (· · · · ·) NCS

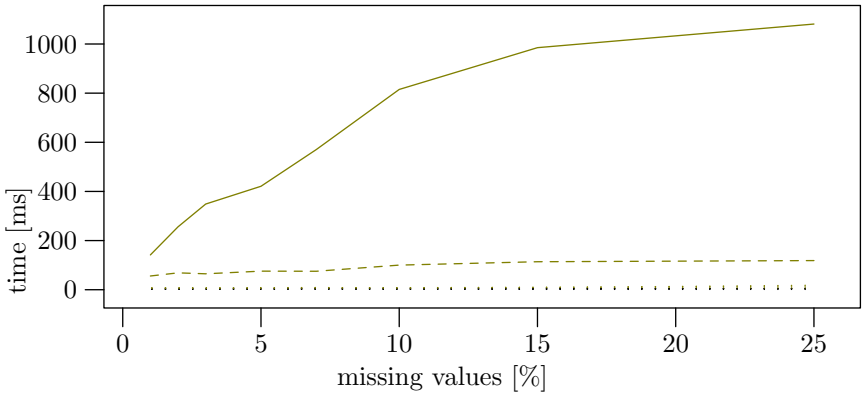

Figure 4: Time of clustering of the 'Iris' data set. The results are gathered in Tab. 2. The results for the NCS algorithm are drawn with the solid line, the results for the IFCM with the dashed line. All other approaches are marked with dots.

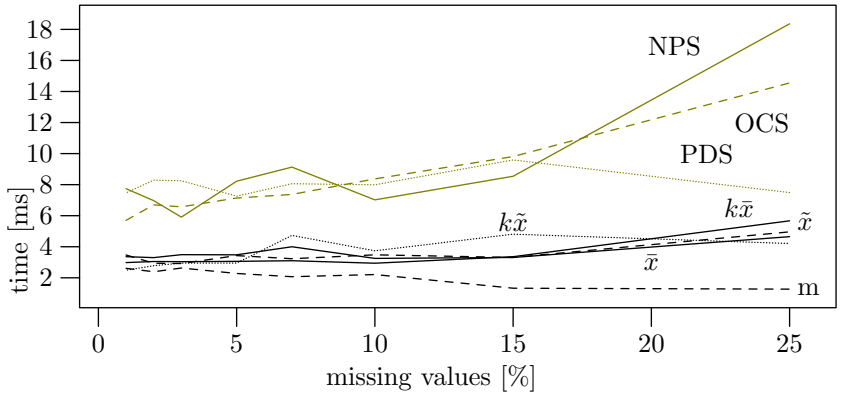

Figure 5: The same as Fig. 4, but without NCS and IFCM. The results are gathered in Tab. 2. Abbreviations: 'm' – marginalisation of objects (dashed line), '$\bar{x}$' – average imputation (solid line), '$\tilde{x}$' – median imputation (dashed lined), '$k\bar{x}$' – kNN average imputation (solid line), '$k\tilde{x}$' – kNN median imputation (dots).

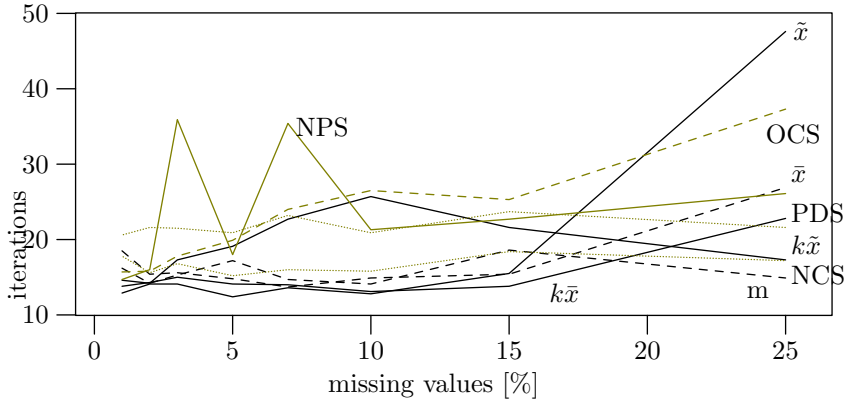

Figure 6: Iterations of various clustering approaches for the 'Iris' data set. The results are gathered in Tab. 2. Abbreviations: 'm' – marginalisation of objects (dashed line), '$\bar{x}$' – average imputation (dashed line), '$\tilde{x}$' – median imputation (solid line), '$k\bar{x}$' – kNN average imputation (solid line), '$k\tilde{x}$' – kNN median imputation (solid line), PDS (dots), OCS (dashed line), NPS (solid line) and NCS (dots).

Table 4: The dissimilarity index between cluster set for the complete 'Iris' data set and incomplete sets with missing values from one cluster.

		FCM				specialised algorithms				
				k-nn	k-nn					
%	mar	$\bar{x}$	$\tilde{x}$	$\bar{x}$	$\tilde{x}$	IFCM	PDS	OCS	NPS	NCS
1	0.01	0.04	0.04	0.03	0.03	0.31	0.13	0.11	0.11	0.11
2	0.01	0.05	0.05	0.18	0.17	0.31	0.13	0.11	0.11	0.11
5	0.31	0.07	0.05	0.04	0.04	0.33	0.13	0.12	0.12	0.12
7	0.31	0.08	0.06	0.05	0.05	0.03	0.14	0.12	0.14	0.12
10	0.61	0.09	0.07	0.07	0.06	0.07	0.14	0.13	0.13	0.13
15	0.19	0.13	0.10	0.08	0.08	0.07	0.15	0.13	0.15	0.13
25	0.37	0.21	0.16	0.13	0.14	0.13	0.15	0.13	0.15	0.14
50	0.66	0.33	0.23	0.20	0.22	0.21	0.19	0.14	0.19	0.28
75	0.7	0.43	0.25	0.26	0.28	0.22	0.54	0.26	0.25	0.49

Table 5: The dissimilarity index between cluster sets for the complete 'Iris' data set and incomplete sets with missing values from one attribute.

		FCM				specialised algorithms				
				k-nn	k-nn					
%	mar	$\bar{x}$	$\tilde{x}$	$\bar{x}$	$\tilde{x}$	IFCM	PDS	OCS	NPS	NCS
1	0.15	0.32	0.46	0.33	0.03	0.01	0.13	0.11	0.11	0.12
2	0.46	0.04	0.04	0.04	0.47	0.16	0.13	0.11	0.11	0.12
5	0.61	0.06	0.05	0.05	0.32	0.29	0.13	0.12	0.11	0.12
7	0.17	0.07	0.06	0.06	0.05	0.16	0.13	0.12	0.12	0.12
10	0.17	0.09	0.09	0.08	0.06	0.04	0.13	0.12	0.11	0.12
15	0.32	0.12	0.12	0.11	0.08	0.05	0.13	0.12	0.12	0.13
25	0.34	0.16	0.16	0.15	0.12	0.44	0.13	0.13	0.12	0.14
50	0.38	0.35	0.35	0.33	0.29	0.56	0.13	0.15	0.14	0.16
75	0.42	0.70	0.76	0.62	0.55	1.22	0.13	0.17	0.14	0.20

values from each cluster. The results are very similar, so we present them only once. Very similar situation was detected for data sets with missing values from one attribute (Tab. 5). Although the attributes have different meanings, the results for values missing form various attributes are very similar.

Table 6: The partition coefficient (Eq. 8) for clusters elaborated by various methods for the 'Iris' data set.

	FCM					specialised algorithms				
				k-nn	k-nn					
%	mar	$\bar{x}$	$\tilde{x}$	$\bar{x}$	$\tilde{x}$	IFCM	PDS	OCS	NPS	NCS
1	−0.85	−0.84	−0.84	−0.84	−0.84	−0.85	−0.78	−0.63	−0.63	−0.63
2	−0.85	−0.82	−0.82	−0.81	−0.82	−0.84	−0.78	−0.63	−0.63	−0.63
3	−0.85	−0.80	−0.80	−0.80	−0.80	−0.84	−0.78	−0.62	−0.63	−0.63
5	−0.85	−0.77	−0.77	−0.77	−0.77	−0.84	−0.78	−0.62	−0.62	−0.62
7	−0.85	−0.74	−0.74	−0.74	−0.74	−0.84	−0.78	−0.62	−0.62	−0.62
10	−0.84	−0.69	−0.68	−0.68	−0.69	−0.84	−0.79	−0.62	−0.62	−0.62
15	−0.85	−0.60	−0.59	−0.60	−0.62	−0.81	−0.78	−0.59	−0.61	−0.59
25	−0.86	−0.47	−0.45	−0.48	−0.50	−0.84	−0.78	−0.59	−0.61	−0.57

Table 7: The partition entropy (Eq. 9) for clusters elaborated by various methods for the 'Iris' data set.

	FCM					specialised algorithms				
				k-nn	k-nn					
%	mar	$\bar{x}$	$\tilde{x}$	$\bar{x}$	$\tilde{x}$	IFCM	PDS	OCS	NPS	NCS
1	0.27	0.30	0.30	0.30	0.30	0.27	0.40	0.65	0.65	0.65
2	0.26	0.32	0.32	0.32	0.32	0.28	0.40	0.65	0.65	0.65
3	0.28	0.35	0.35	0.35	0.35	0.29	0.40	0.66	0.65	0.65
5	0.26	0.40	0.40	0.40	0.40	0.29	0.40	0.66	0.66	0.66
7	0.28	0.46	0.46	0.46	0.46	0.29	0.40	0.66	0.66	0.66
10	0.26	0.54	0.54	0.54	0.54	0.31	0.40	0.67	0.66	0.67
15	0.25	0.66	0.67	0.65	0.66	0.33	0.39	0.69	0.66	0.69
25	0.26	0.90	0.94	0.86	0.85	0.34	0.40	0.73	0.66	0.76

Table 8: The Xie-Beni index (Eq. 10) for clusters elaborated by various methods for the 'Iris' data set.

	FCM					specialised algorithms			
				k-nn	k-nn				
%	mar	$\bar{x}$	$\tilde{x}$	$\bar{x}$	$\tilde{x}$	IFCM	PDS	OCS	NPS
1	0.36	0.38	0.27	0.27	0.27	0.26	0.29	0.29	0.29
2	0.25	0.29	0.29	0.29	0.29	0.47	0.30	0.29	0.29
3	0.48	0.30	0.31	0.32	0.31	0.27	0.30	0.29	0.29
5	0.25	0.33	0.35	0.35	0.31	0.26	0.32	0.32	0.30
7	0.37	0.38	0.40	0.40	0.35	0.29	0.33	0.32	0.32
10	0.36	0.48	0.51	0.50	0.39	0.30	0.33	0.31	0.31
15	0.35	0.52	0.60	0.61	0.33	0.28	0.35	0.34	0.35
25	0.34	22.97	147	1.74	0.62	0.29	0.38	0.37	0.37

Table 9: The Fukuyama-Sugeno index (Eq. 11) for clusters elaborated by various methods for the 'Iris' data set.

	FCM					specialised algorithms			
%	mar	$\bar{x}$	$\tilde{x}$	k-nn $\bar{x}$	k-nn $\tilde{x}$	IFCM	PDS	OCS	NPS
1	−71.31	−70.56	−72.81	−72.94	−72.95	−75.20	−53.86	−54.61	−54.61
2	−72.81	−70.80	−71.04	−71.45	−71.56	−75.08	−54.35	−54.80	−54.60
3	−68.90	−66.35	−66.57	−67.02	−66.61	−74.70	−53.15	−54.63	−53.98
5	−59.19	−58.53	−58.81	−59.97	−57.18	−71.56	−51.18	−53.92	−53.35
7	−54.71	−49.68	−50.22	−52.16	−49.19	−68.89	−49.58	−54.14	−52.62
10	−50.72	−40.21	−41.66	−43.42	−41.14	−70.80	−52.48	−54.32	−50.68
15	−40.05	−22.52	−22.61	−27.65	−24.18	−63.71	−46.33	−52.54	−45.41
25	−24.19	7.94	14.48	0.54	1.89	−61.57	−23.82	−52.55	−33.31

Table 10: The Bensaid index (Eq. 12) for clusters elaborated by various methods for the 'Iris' data set.

	FCM					specialised algorithms			
%	mar	$\bar{x}$	$\tilde{x}$	k-nn $\bar{x}$	k-nn $\tilde{x}$	IFCM	PDS	OCS	NPS
1	0.19	0.22	0.21	0.22	0.22	0.19	0.17	0.17	0.17
2	0.18	0.22	0.23	0.22	0.23	0.20	0.17	0.17	0.17
3	0.20	0.26	0.25	0.26	0.26	0.21	0.18	0.17	0.17
5	0.19	0.29	0.28	0.28	0.31	0.23	0.18	0.17	0.18
7	0.20	0.32	0.30	0.30	0.41	0.24	0.19	0.17	0.18
10	0.19	0.40	0.36	0.37	0.45	0.24	0.19	0.17	0.20
15	0.19	0.51	0.44	0.44	0.61	0.28	0.20	0.17	0.22
25	0.18	0.81	0.83	0.64	2.91	0.34	0.19	0.17	0.29

Table 11: The Czogała-Łęski index (Eq. 13) for clusters elaborated by various methods for the 'Iris' data set.

	FCM					specialised algorithms			
%	mar	$\bar{x}$	$\tilde{x}$	k-nn $\bar{x}$	k-nn $\tilde{x}$	IFCM	PDS	OCS	NPS
1	0.0037	0.0037	0.0038	0.0037	0.0039	0.0036	0.0046	0.0046	0.0045
2	0.0039	0.0042	0.0042	0.0041	0.0042	0.0037	0.0047	0.0045	0.0046
3	0.0041	0.0045	0.0045	0.0045	0.0045	0.0037	0.0047	0.0046	0.0047
5	0.0045	0.0053	0.0053	0.0052	0.0056	0.0044	0.0049	0.0046	0.0047
7	0.0045	0.0060	0.0060	0.0060	0.0068	0.0043	0.0051	0.0048	0.0049
10	0.0052	0.0075	0.0076	0.0073	0.0099	0.0042	0.0050	0.0047	0.0050
15	0.0070	0.0115	0.0115	0.0109	0.0128	0.0049	0.0056	0.0049	0.0057
25	0.0132	0.0260	0.0376	0.0224	0.0238	0.0057	0.0069	0.0052	0.0079

Table 12: The dissimilarity index between clusters elaborated for the complete and incomplete 'Telugu' data set.

	FCM					specialised algorithms		
				k-nn	k-nn			
%	mar	$\bar{x}$	$\tilde{x}$	$\bar{x}$	$\tilde{x}$	IFCM	OCS	PDS
1	0.6530	0.2691	1.2504	0.7595	1.0679	0.8654	0.6266	1.0098
2	0.7329	0.5791	0.3620	0.3607	0.6678	0.4051	0.6383	0.5646
5	0.7783	0.4458	0.7408	0.6743	0.4271	0.8004	1.1059	0.8823
10	0.7614	0.5916	0.5635	0.6538	0.6015	0.6879	0.3507	0.3498
20	0.6659	0.5250	0.5755	0.6122	0.4861	0.9233	0.6254	0.2814
50	0.3613	1.0112	1.0260	1.0258	1.0602	4.8977	1.4857	0.3196

Table 13: The partition coefficient (Eq. 8) for clusters elaborated by various methods for the 'Telugu' data set.

	FCM					specialised algorithms		
				k-nn	k-nn			
%	mar	$\bar{x}$	$\tilde{x}$	$\bar{x}$	$\tilde{x}$	IFCM	OCS	PDS
1	−0.7529	−0.7376	−0.7389	−0.7389	−0.7389	−0.7519	−0.4794	−0.5487
2	−0.7526	−0.7222	−0.7276	−0.7273	−0.7269	−0.7520	−0.4808	−0.5517
5	−0.7501	−0.6858	−0.6887	−0.6885	−0.6848	−0.6711	−0.4804	−0.5504
10	−0.7574	−0.5984	−0.5997	−0.5997	−0.5997	−0.7645	−0.4870	−0.5518
20	−0.7570	−0.4280	−0.4287	−0.4283	−0.4219	−0.6973	−0.4919	−0.5606
50	−0.7857	−0.1667	−0.1667	−0.1667	−0.1667	−0.6592	−0.5513	−0.6332

Table 14: The Xie-Beni (Eq. 10) for clusters elaborated by various methods for the 'Telugu' data set.

	FCM					specialised algorithms		
				k-nn	k-nn			
%	mar	$\bar{x}$	$\tilde{x}$	$\bar{x}$	$\tilde{x}$	IFCM	OCS	PDS
1	0.2292	0.2321	0.2303	0.2304	0.2387	0.2978	0.2228	2.9053
2	0.2345	0.2655	0.2292	0.2290	0.2324	0.4033	0.2308	3.4253
5	0.2302	0.2725	0.3019	0.3014	0.2791	1.7502	0.2308	4.9126
10	0.2231	0.6452	0.6729	0.6730	0.6946	0.5748	0.2008	8.4114
20	0.2169	73.9845	$2.92 \cdot 10^7$	$1.39 \cdot 10^7$	$4.95 \cdot 10^4$	2.3041	0.1768	56.9151
50	0.2202	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$1.54 \cdot 10^9$	0.1616	74.0118

Table 15: The Fukuyama-Sugeno (Eq. 11) for clusters elaborated by various methods for the 'Telugu' data set.

	FCM					specialised algorithms		
				k-nn	k-nn			
%	mar	$\bar{x}$	$\tilde{x}$	$\bar{x}$	$\tilde{x}$	IFCM	OCS	PDS
1	−219	−216	−216	−216	−216	−225	−140	−135
2	−215	−207	−209	−209	−209	−219	−142	−123
5	−195	−181	−181	−181	−180	−196	−143	−95
10	−165	−128	−128	−128	−128	−255	−146	−42
20	−115	−40	−40	−40	−42	−236	−155	38
50	−29	92	90	90	87	−322	−182	187

Table 16: The Bensaid index (Eq. 12) for clusters elaborated by various methods for the 'Telugu' data set.

	FCM					specialised algorithms		
				k-nn	k-nn			
%	mar	$\bar{x}$	$\tilde{x}$	$\bar{x}$	$\tilde{x}$	IFCM	OCS	PDS
1	0.1250	0.1437	0.1499	0.1500	0.1581	0.1384	0.0902	0.1739
2	0.1234	0.1754	0.1744	0.1736	0.1724	0.1741	0.0868	0.2237
5	0.1222	0.2815	0.4727	0.4582	0.3436	0.2425	0.0865	0.3094
10	0.1288	0.7057	2.0071	2.0053	0.8223	23.5907	0.0912	0.4934
20	0.1222	7.5756	3.4237	3.2467	1.2089	0.3507	0.0937	0.7631
50	0.1207	$2.24 \cdot 10^{13}$	17.5297	13.2160	2.6080	0.0819	0.0929	1.4597

Table 17: The Czogała-Łęski (Eq. 13) for clusters elaborated by various methods for the 'Telugu' data set.

	FCM					specialised algorithms		
				k-nn	k-nn			
%	mar	$\bar{x}$	$\tilde{x}$	$\bar{x}$	$\tilde{x}$	IFCM	OCS	PDS
1	0.0007	0.0008	0.0008	0.0008	0.0008	0.0008	0.0009	0.0017
2	0.0007	0.0009	0.0009	0.0009	0.0009	0.0010	0.0009	0.0022
5	0.0008	0.0014	0.0014	0.0014	0.0014	0.0014	0.0009	0.0031
10	0.0010	0.0028	0.0028	0.0028	0.0028	0.0013	0.0009	0.0049
20	0.0014	0.0100	0.0099	0.0099	0.0110	0.0020	0.0009	0.0076
50	0.0049	$3.86 \cdot 10^{12}$	$4.12 \cdot 10^{12}$	$5.03 \cdot 10^{12}$	$3.98 \cdot 10^{13}$	0.0007	0.0008	0.0144

6 Conclusions

The paper presents the comparison of several clustering algorithms for data sets with missing values. We analysed the preprocessing techniques and specialised algorithms for data missing values. The experiments show that for moderate missing ratio ($< 10\%$) it is more advantageous to use preprocessing method (the best is median imputation). For high missing ratios ($> 25\%$) the specialised algorithms should be used, but it is worth mentioning that simple marginalisation can elaborate better results than imputation in preprocessing. The results seems to be independent of the type of data loss: missing from the whole data set, missing from only one cluster or attribute. The proposed dissimilarity measure (although its different objective) seems to be in concordance with the cluster quality indices.

Acknowledgements

The authors are grateful to the anonymous referees for their constructive comments that have helped to improve the paper.

References

[1] Acuña E., Rodriguez C., The treatment of missing values and its effect in the classifier accuracy. In Banks D., House L., McMorris F. R., Arabie P., Gaul W. (eds.), editors, *Classification, Clustering and Data Mining Applications, Springer, Berlin, Heidelberg*, 2004, 639–648.

[2] Alcalá-Fdez J., Fernandez A., Luengo J., Derrac J., García S., Sánchez L., Herrera F., KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, **17**, 2-3, 2011, 255–287.

[3] Bensaid A. M., Hall L. O., Bezdek J. C., Clarke L. P., Silbiger M. L., Arrington J. A., R. F. Murtagh, Validity-guided (re)clustering with applications to image segmentation. *Transactions on Fuzzy Systems*, **4**, 2, 1996, 112–123.

[4] Chan L., Gilman J., Dunn O., Alternative approaches to missing values in discriminant analysis. *Journal of the American Statistical Association*, **71**, 356, 1976, 842–844.

[5] Czogała E., Łęski J., *Fuzzy and Neuro-Fuzzy Intelligent Systems*. Series in Fuzziness and Soft Computing. Physica-Verlag, A Springer-Verlag Company, Heidelberg, New York, 2000.

[6] Dempster A. P., Laird N. M., Rubin D. B., Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, **39**, 1, 1977, 1–38.

[7] Dunn J. C., A fuzzy relative of the ISODATA process and its use in detecting compact, well separated clusters. *Journal Cybernetics*, **3**, 3, 1973, 32–57.

[8] Ghahramani Z, Jordan M. I., Learning from incomplete data. Technical report, Lab Memo No. 1509, CBCL Paper No. 108, MIT AI Lab, 1995.

[9] Grzymała-Busse J., Hu M., A comparison of several approaches to missing attribute values in data mining. In Ziarko W. and Yao Y. (eds), *Rough Sets and Current Trends in Computing*, volume 2005 of *Lecture Notes in Computer Science*, 378–385. Springer Berlin / Heidelberg, 2001.

[10] Grzymała-Busse J., Grzymała-Busse W., Handling missing attribute values. In Maimon O., Rokach L. (eds), *The Data Mining and Knowledge Discovery Handbook*, 37–57. Springer, 2005.

[11] Grzymala-Busse J., Siddhaye S., Rough set approaches to rule induction from incomplete data. In *Proceedings of the IPMU'2004, the 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, volume 2, pages 923–930, 2004.

[12] Gustafson D., Kessel W., Fuzzy clustering with a fuzzy covariance matrix. In *Decision and Control including the 17th Symposium on Adaptive Processes, 1978 IEEE Conference on*, volume 17, pages 761–766, 1978.

[13] Hathaway R. J., Bezdek J. C., Fuzzy c-means clustering of incomplete data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, **31**, 5, 2001, 735–744.

[14] Himmelspach L., Conrad S., Fuzzy clustering of incomplete data based on cluster dispersion. In Hüllermeier E., Kruse R., Hoffmann F. (eds), *Computational Intelligence for Knowledge-Based Systems Design*, volume 6178 of *Lecture Notes in Computer Science*, pages 59–68. Springer Berlin / Heidelberg, 2010.

[15] Little R. J., Rubin D. B., *Statistical analysis with missing data*. John Wiley and Sons, New York, 1987.

[16] Pal N. R., Bezdek J. C., On cluster validity for the fuzzy c-means model. *IEEE Transactions on Fuzzy Systems*, **3**, 3, 1995, 370–379.

[17] Pal S. K., Majumder D. D., Fuzzy sets and decision making approaches in vowel and speaker recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, **7**, 1977, 625–629.

[18] Renz C., Rajapakse J., Razvi K., Liang S., Ovarian cancer classification with missing data. In *Proceedings of the 9th International Conference on Neural Information Processing, ICONIP'02*, volume 2, pages 809–813, Singapore, 2002.

[19] Simiński K., Neuro-rough-fuzzy approach for regression modelling from missing data. *International Journal of Applied Mathematics and Computer Science*, **22**, 2, 2012, 461–476.

[20] Simiński K., Clustering with missing values. *Fundamenta Informaticae*, **123**, 3, 2013, 331–350.

[21] Timm H., Kruse R., Fuzzy cluster analysis with missing values. In *NAFIPS 1998 Conference of the North American Fuzzy Information Processing Society*, pages 242–246, 1998.

[22] Troyanskaya O., Cantor M., Sherlock G., Brown P., Hastie T., Tibshirani R., Botstein D., Altman R., Missing value estimation methods for DNA microarrays. *Bioinformatics*, **17**, 6, 2001, 520–525.

[23] Wagstaff K., Laidler V., Making the most of missing values: Object clustering with partial data in astronomy. In *Proceedings of Astronomical Data Analysis Software and Systems XIV*, volume 347, pages 172–176, Pasadena, California, USA, 2005.

[24] Xie X., Beni G., A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**, 8, 1991, 841–847.

[25] Yao L., Weng K., Chang R., Fuzzy classification of incomplete data with adaptive volume. In *ACIIDS '09: Proceedings of the 2009 First Asian Conference on Intelligent Information and Database Systems*, pages 232–237, Washington, DC, USA, 2009.

[26] Zhang C., Zhu X., Zhang J., Qin Y., Zhang S., GBKII: An imputation method for missing values. *Advances in Knowledge Discovery and Data Mining*, **4426**, 2007, 1080–1087.

[27] Zhang S., Shell-neighbor method and its application in missing data imputation. *Applied Intelligence*, **35**, 1, 2011, 1–11.

Received May,2013