Napsu Karmitsa | Sona Taheri | Adil Bagirov | Pauliina Mäkinen

Clusterwise Linear Regression based Missing Value Imputation for Data Pre-processing

Turku Centre *for* Computer Science

TUCS Technical Report
No 1193, March 2018

Clusterwise Linear Regression based Missing Value Imputation for Data Preprocessing

Napsu Karmitsa
Department of Mathematics and Statistics
University of Turku
FI-20014 Turku, Finland
napsu@karmitsa.fi

Sona Taheri
Faculty of Science and Technology,
Federation University Australia,
Victoria, Australia
staheri@federation.edu.au

Adil Bagirov
Faculty of Science and Technology,
Federation University Australia,
Victoria, Australia
a.bagirov@federation.edu.au

Pauliina Mäkinen
Department of Mathematics and Statistics
University of Turku
FI-20014 Turku, Finland
kjjoki@utu.fi

TUCS Technical Report
No 1193, March 2018

Abstract

We introduce a new accurate method for preprocessing incomplete data sets. We combine two well-known approaches for missing value imputation: the linear regression and the clustering. That is, we use the clusterwise linear regression to predict suitable imputations. A clusterwise linear regression problem consists of finding a number of linear functions each approximating a subset of the given data. The idea here is to approximate missing values using only those data points that are somewhat similar to the incomplete data object. This idea is used also in clustering based imputations. On the other hand, we use linear regression within the given cluster to find accurate predictions to the missing values and we do this simultaneously to clustering. The aim here is to make an accurate and efficient method for preprocessing incomplete data sets. The proposed algorithm is tested on small and large, artificial and real world data sets and compared with other algorithms for missing data imputation. Numerical results demonstrate that the proposed algorithm produces the most accurate imputations in data sets with clear structure and small or moderate amount of missing values.

Keywords: Data analysisi; Incomplete data; Imputation; Clusterwise linear regression; Nonsmooth Optimization.

TUCS Laboratory
Turku Optimization Group (TOpGroup)

1 Introduction

The occurrence of missing (or incomplete) data is very common in many fields of research such as social sciences, biology, medicine and climatic science. There are various reasons for possible incompleteness of data. For instance, in medical domain some data may be missing because certain procedures were not performed on a given patient, other data may be missing because the patient chose not to disclose them, and even some data may be missing due to malfunction of certain equipment [25].

As the quality of knowledge extracted from data depends largely on the quality of data, missing values may have a significant effect on the conclusions that can be drawn from the data. Moreover, most of the existing knowledge discovery and data mining algorithms, used, for example, for clustering and classification, are designed under the assumption that there are no missing values in the data. When data is incomplete, the performance of these algorithms may worsen drastically or they may not work at all. Thus, data pre-processing is a critical task in the knowledge discovery process in order to ensure the quality of the data to be analyzed and the performance of the tools to be used.

When dealing with incomplete data, possible approaches can be divided into three main categories:

1. deletion-based methods,

2. learning methods for complete and incomplete data, and

3. imputation methods.

The deletion-based methods (e.g. pairwise deletion and listwise deletion [16, 32, 55]) strive toward complete database by removing all the observations/attributes containing missing values. Because of their simplicity, these methods are fairly popular. However, the methods may lead to large losses of information, which calls for thorough consideration before using them [16].

The second approaches, learning methods, apply machine-learning techniques to classify or cluster incomplete data directly without explicitly estimating missing features or modifying the data set. Examples of these kind of methods include clustering based methods like modifications of k-means [12, 50] and fuzzy c-means [24, 58], neural network based approaches [17, 19, 37, 42, 56] and different variants of kernel methods [27, 33, 47].

The third approaches, imputation methods, fill missing values in order to complete the original database without significant loss of information. The key advantage of these methods is the ability to create complete data set by embedding new values (predictions) without changing the original observed values in the database. This new imputed data set can then be treated with any traditional data mining method for complete data.

On their turn, imputation methods can be divided into three groups including

1. data driven,

2. model-based, and

3. machine learning based approaches.

Data driven imputation methods usually produce the imputed values by relatively simple statistical/mathematical methods like mean, conditional mean, hot-deck, cold-deck, or substitution [1, 16].

Model-based imputation methods use mathematical or statistical models to handle the missing values in the data and to predict correct imputations. This group consists mainly of regression-based and maximum likelihood based approaches like multiple imputations by chained equations and stochastic regression [16, 40, 41, 53], and expectation-maximization (EM) [16, 51, 54].

Various machine learning based approaches have been proposed for missing value imputation. These include neural network based approaches [21, 44], clustering based approaches [30, 36, 9], and K-nearest neighbours (K-nn) [8, 48] to mention but few. In addition, imputation approaches that combine machine learning techniques and model based approaches are introduced, for instance in [43, 57] where clustering is used together with the linear regression. This is the case also in this paper, where we introduce a new imputation algorithm IVIACLR (*Imputation via Clusterwise Linear Regression*).

The clusterwise regression is a technique to approximate data using two or more regression functions. It is based on two well-known techniques: clustering and regression, and simultaneously identifies clusters and their associated regression functions. If the regression functions are linear then the clusterwise regression is called the *clusterwise linear regression* (CLR). The CLR has many applications (see, e.g. [4, 26, 35, 38, 39, 52]). Here, we will use it as the part of a new imputation algorithm IVIACLR. The main idea in the IVIACLR is to use regression of those data points that are somewhat similar to the incomplete data object. That is, we infer the value of a missing feature based on that item's observed features and its similarity to other items in the data set. The difference of the IVIACLR to the methods introduced in [43, 57] is that, instead of using clustering and regression separately as in [43, 57], we use the CLR to predict suitable imputations. Thus, the IVIACLR computes all the predictions to missing values simultaneously to clustering and instead of the traditional ball-shaped clusters with cluster centres our clusters are regression functions.

The algorithms for solving general CLR problems can be divided roughly into three groups: algorithms which are based on data mining [18, 45, 46]; statistical algorithms [14, 20, 39]; and optimization based methods [5, 6, 7, 10, 11, 15, 28]. In principle, any of these approaches could be used in our new imputation method IVIACLR. In this paper, we have adopted the nonsmooth nonconvex optimization formulation of the CLR problem and apply the algorithm LMBM-CLR [28] to solve it.

The LMBM-CLR -method consists of two algorithms: an incremental algorithm [34] is used to solve CLR problems globally and at each iteration of this algorithm

the limited memory bundle algorithm (LMBM) [22, 23] is used to solve both the CLR problem and the so-called auxiliary CLR problem with different starting points provided by the incremental algorithm. In addition, we have added here three different kind of prediction approaches to choose best possible predictions to missing values.

The performance of the proposed method IVIACLR is studied and compared to other imputation methods on three artificial and five real data sets of various sizes with varying percentages of missing values. The evaluation criteria used in our experiments are the root mean square error (RMSE), mean absolute error (MAE), and unsupervised classification error (UCE). In addition, we introduce a new *cluster center misplacement* (CCM) criterion that can be used together with the UCE to measure the bias in the imputed values.

The rest of this paper is organized as follows. In the next section, we give some background information, including the nonsmooth optimization formulation of the CLR problem and basic ideas of the LMBM-CLR. Section 3 introduces the new imputation method and the prediction approaches used. The results of the numerical experiments and the new evaluation criterion CCM are presented and discussed in Section 4, and finally, Section 5 concludes the paper.

2 Background

2.1 Notations and Definitions

Throughout the paper the following notations are used: the Euclidean norm in $\mathbb{R}^n$ is denoted by $\|\cdot\|$ and the inner product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is denoted by $\boldsymbol{a}^T\boldsymbol{b}$ (bolded symbols are used for vectors).

We have a data set $A = \{\hat{\boldsymbol{a}}_1, \hat{\boldsymbol{a}}_2, \ldots \hat{\boldsymbol{a}}_m\}$ of m objects (observations, instances, data points) and each object has n features (attributes, variables). We denote by $\hat{a}_{ij}$ ($1 \leq i \leq m$ and $1 \leq j \leq n$) the value of the attribute j in object $\hat{\boldsymbol{a}}_i$. Data point $\hat{\boldsymbol{a}}_i$ is called *complete*, if $\hat{a}_{ij} \neq \emptyset$ with all $j = 1, \ldots, n$, and *incomplete*, if $\hat{a}_{ij} = \emptyset$ with at least one $j \in \{1, \ldots, n\}$. In the latter case, we say that object $\hat{\boldsymbol{a}}_i$ has a *missing value* on attribute j. The attributes $\hat{a}_{ij}$, $j \in \{1, \ldots, n\}$ that are available for an incomplete object $\hat{\boldsymbol{a}}_i$ are called the *reference attributes*. Our objective is to find and impute the values of non-reference attributes for incomplete objects.

For regression purposes we denote $A = \{(\boldsymbol{a}_1, b_1), (\boldsymbol{a}_2, b_2), \ldots, (\boldsymbol{a}_m, b_m)\}$, where $\boldsymbol{a}_i \in \mathbb{R}^{n-1}$ ($i = 1, \ldots, m$) are the so-called *input variables* and $b_i \in \mathbb{R}$ is the *output variable*. If there are missing values only in one feature, say in feature j of the data set A, then we set $b_i = \hat{a}_{ij}$, $i = 1, \ldots, m$, and the rest of the variables are input variables. However, in real world data sets it is common for missing values to occur in several variables. In such situations, we go trough all features with missing values iteratively using some initial imputations in place of missing values on those features that are not output variables in the current iteration.

2.2 Nonsmooth Optimization

Nonsmooth optimization (NSO) refers to the general problem of minimizing (or maximizing) functions that are typically not differentiable at their minimizers (maximizers). In NSO the gradient $\nabla f(\boldsymbol{x})$ needs not to exist for all $\boldsymbol{x} \in \mathbb{R}^n$. However, we can define the so-called subdifferential [13] that allows us to generalize the classical theory of optimization to NSO. The *subdifferential* $\partial f(\boldsymbol{x})$ of a locally Lipschitz continuous function f is given by

$$\partial f(\boldsymbol{x}) = \mathrm{conv}\{ \lim_{i \to \infty} \nabla f(\boldsymbol{x}_i) \mid \boldsymbol{x}_i \to \boldsymbol{x} \text{ and } \nabla f(\boldsymbol{x}_i) \text{ exists} \},$$

where "conv" denotes the convex hull of a set. Each component $\boldsymbol{\xi} \in \partial f(\boldsymbol{x})$ is called a *subgradient* of f at $\boldsymbol{x}$. For more details on nonsmooth analysis and optimization, we refer to [3].

2.3 Clusterwise Linear Regression

The aim of the CLR is to find an optimal partition of the given data set $A = \{(\boldsymbol{a}_i, b_i) \in \mathbb{R}^{n-1} \times \mathbb{R} \mid \ i = 1, \ldots, m\}$ into k clusters and, simultaneously, to find regression coefficients $\{\boldsymbol{x}_j, y_j\},\ \boldsymbol{x}_j \in \mathbb{R}^{n-1},\ y_j \in \mathbb{R},\ j = 1, \ldots, k$ within clusters in order to minimize the overall fit. Let $A^j \subset A,\ j = 1, \ldots, k$ be clusters such that

1. $A^j \neq \emptyset$, $j = 1, \ldots, k$;
2. $A^j \bigcap A^l = \emptyset$, for all $j, l = 1, \ldots, k$, $j \neq l$;
3. $A = \bigcup_{j=1}^{k} A^j$.

Let $\{\boldsymbol{x}_j, y_j\}$ be linear regression coefficients computed using solely the data points from the cluster $A^j,\ j = 1, \ldots, k$. Then for a given data point $(\boldsymbol{a}, b) \in A$ and coefficients $\{\boldsymbol{x}_j, y_j\}$ the squared regression error $E_{\boldsymbol{a}b}(\boldsymbol{x}_j, y_j)$ is given by

$$E_{\boldsymbol{a}b}(\boldsymbol{x}_j, y_j) = \left((\boldsymbol{x}_j)^T \boldsymbol{a} + y_j - b\right)^2.$$

A data point is associated with the cluster whose regression error at this point is the smallest one. The function

$$f_k(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{m} \min_{j=1,\ldots,k} E_{\boldsymbol{a}b}(\boldsymbol{x}_j, y_j),$$

is called the *k-th clusterwise linear regression function* or the *k-th overall fit function* [5, 6, 7]. Here $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) \in \mathbb{R}^{(n-1)k}$ and $\boldsymbol{y} = (y_1, \ldots, y_k) \in \mathbb{R}^k$. The *NSO formulation of the CLR problem* is given by

$$\begin{cases} \text{minimize} & f_k(\boldsymbol{x}, \boldsymbol{y}) \\ \text{subject to} & \boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) \in \mathbb{R}^{(n-1)k},\ \boldsymbol{y} \in \mathbb{R}^k. \end{cases} \tag{1}$$

For $k = 1$ Problem (1) is convex and for $k > 1$ it is nonsmooth, nonconvex, and piecewise quadratic. The number of clusters k is not always known a priori and this number should be specified before solving Problem (1). The number of variables in Problem (1) is $n \times k$ and it does not depend on m, the number of points in a data set.

2.4 LMBM-CLR -Method

In this subsection we recall the structure of the LMBM-CLR -method for solving CLR problems. As already said in the introduction, the LMBM-CLR -method consists of two algorithms: an incremental algorithm is used to solve CLR problems globally and at each iteration of this algorithm the LMBM is used to solve the CLR problem (1) and the so-called *auxiliary CLR problem* that is used to find good initial solutions for the CLR problem. Figure 1 illustrates the structure of this combination and basic ideas of the two algorithms. For more details we refer to [28] for the LMBM-CLR, [5] for the incremental algorithm and auxiliary problem, and [22, 23] for the basic LMBM.

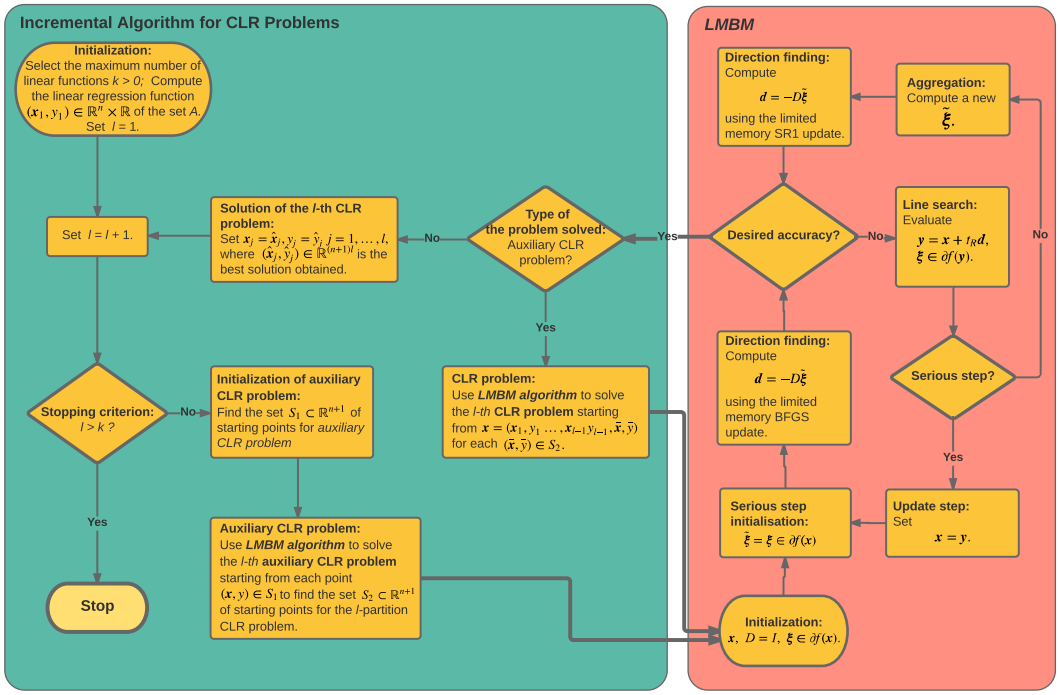


Figure 1: LMBM-CLR -method. Here, D denotes the inverse variable metric approximation of the Hessian and $\boldsymbol{\xi} \in \partial f(\boldsymbol{x})$ is an arbitrary component (the so-called *subgradient*) from the subdifferential.

In addition to the k-CLR problem, the LMBM-CLR solves also all intermediate l-CLR problems, where $l = 1, \ldots, k-1$.

3 Missing Value Imputation via CLR

In this section, we introduce a new imputation method IVIACLR. The IVIACLR consists of three different parts that, at least in principle, can be altered: initial imputation, CLR-method, and predictions. We already introduced the used CLR-method LMBM-CLR in the previous section, so here, we first introduce the main algorithm and then the initial imputations and prediction approaches used.

3.1 Main Algorithm

We recall that we denote $A = \{(\boldsymbol{a}_1, b_1), (\boldsymbol{a}_2, b_2), \ldots, (\boldsymbol{a}_m, b_m)\}$, where $\boldsymbol{a}_i \in \mathbb{R}^{n-1}$ and $b_i \in \mathbb{R}$ are the input and the output variables, respectively ($i = 1, \ldots, m$). The IVIACLR consists of inner and outer iterations. In the inner iteration, we go trough all features with missing values iteratively using some initial imputations (or previously imputed values) in place of missing values on those features that are not output variables in the current iteration. After imputing all the missing values, we repeat the process (outer iteration) with imputed values as place holders until the results are not changing a lot or the maximum number of outer iterations is reached. The IVIACLR -algorithm is as follows:

Algorithm 1: IVIACLR

Data: Incomplete data set $A = \{\hat{\boldsymbol{a}}_1, \hat{\boldsymbol{a}}_2, \ldots \hat{\boldsymbol{a}}_m\}$, final number of regression functions k, number of outerloops $o_{max} \geq 1$, and tolerance for the change $\varepsilon \geq 0$.
Result: Imputed data set A^{imp}.
Use a simple *imputation method* (e.g. mean) to impute all missing values. These imputations are considered as "place holders";
Set $i_{out} = 0$;
while $i_{out} < o_{max}$ **do**
  Set $i_{out} = i_{out} + 1$;
  while *There are features with missing values* **do**
    Find the feature j with most missing values and set the output variable $b_i = \hat{a}_{ij}$, $i = 1, \ldots, m$. The rest of the variables (with place holders and/or previously imputed values) are input variables;
    Use *clusterwise linear regression* to find predictions to missing values at feature j;
    Replace missing values at feature j with *predictions*. Set those values as "not missing";
  if $i_{out} > 1$ **then**
    Compute the difference d between previous and current imputations;
    if $d < \varepsilon$ **then**
      STOP with current imputed data set.
  Set all originally missing values again as "missing";
STOP with current imputed data set;

REMARK 3.1. The best possible number of regression functions (clusters) is data specific and it should be given as input parameter to the IVIACLR. Nevertheless, it may not be known a priori. The use of the LMBM-CLR gives us a possibility to use some intermediate results and possible procedures for "intelligent" stopping will be studied in the future.

REMARK 3.2. In the current version of the IVIACLR we only use linear regression and consider continuous numeric data. In addition, discrete (integer) data can be considered simple by rounding the final result. Further, it would be possible to generalize our approach to deal with different data types (e.g. binary) if some other regression model was used.

3.2 Initial imputations

There are three different options for initial imputations in the IVIACLR.

1. *Mean imputation.* In the mean imputation the mean of all observed values of the feature with the missing value is used as an impute.

2. *Linear regression imputation.* In the linear regression imputation the data is regressed using the complete data set (i.e. data set produced with deletion). The previously regressed values are not taken in to account but the complete data set used in computations is the same for all features with missing values.

3. *Recursive regression imputation.* In the recursive regression imputation we first regress and impute the feature with the least number of missing values using the complete data. Then we repeat the process for the variable with the next fewest missing values using updated data set with previously imputed values and so on until all missing values have been imputed.

As a sole imputation method, the mean is known to produce imputations with high level of bias by pulling the distribution of the imputed data toward the mean of observed data. The positive point is that the mean imputation is extremely easy to compute and we can compute it even if every object in the data set has missing values. Thus, it is well suited as an initial imputation method. On the other hand, the linear (recursive) regression imputation underestimates the variance of imputed items. However, due to fact that we continue by making more regression functions, this is not a big issue in our method. However, we can not compute initial imputations with the regression model if there is no complete data, that is, if every object has one or more missing values.

3.3 Predictions

The selection of predictions in the CLR is not straightforward. In the IVIACLR we have tested three different weighting based prediction approaches. Here, as before, k

is the number of linear regression functions (clusters), $(\boldsymbol{x}_j, y_j) \in \mathbb{R}^{n-1} \times \mathbb{R}$ are the regression coefficients corresponding to the j-th cluster, $j = 1, \ldots, k$, and $(\boldsymbol{a}_i, b_i)$ is a data point with possible missing value in b_i. For an object $(\boldsymbol{a}_i, b_i) \in \mathbb{R}^n$ $(i = 1, \ldots, m)$ with a missing value in b_i we compute

$$\boldsymbol{z}_j = \boldsymbol{x}_j^T \boldsymbol{a}_i + y_j, \quad j = 1, \ldots, k.$$

The following prediction methods are used in the IVIACLR.

1. *Simple weighting method [4].* In the simple weighting method the weight w_j is computed as $w_j = m_j/m$, where m_j is the number of points in the j-th cluster and m is the total number of the points in data set.

2. *Local weighting method.* In the local weighting method we, instead of computing weight from all m data points, use only $l < m$ nearest neighbours to compute the weight. When the nearest neighbours has been selected the weight w_j is computed as $w_j = l_j/l$, where l_j is the number of nearest neighbour points in the j-th cluster.

3. *RMSE based local weighting.* In the RMSE based local weighting method we first compute how similar the data point $(\boldsymbol{a}_i, b_i)$ with a missing value in b_i is to l of its nearest neighbours $(\boldsymbol{a}_h, b_h)$, $h \in \{1, \ldots, m\}, h \neq i$, using the root mean square error (RMSE). Then we compute the weight w_j as

$$w_j = \sum_{h=\mathcal{C}_j} \frac{\overline{rmse} - rmse_h}{(l-1)\overline{rmse}},$$

   where $\mathcal{C}_j$ is the set of indices of nearest neighbour points in the j-th cluster ($|\mathcal{C}_j| = l_j$), $\overline{rmse} = \sum_{h=1}^{l} rmse_h$, and $rmse_h$ is the RMSE between $\boldsymbol{a}_i$ and $\boldsymbol{a}_h$. Here, we set $w_j = l_j/l$ if $\overline{rmse} = 0$. On the other hand, if $l = 1$ we simply take the nearest neighbour (the one with the smallest $rmse_h$) and the cluster j^* it belongs, and we set $w_{j^*} = 1$ and $w_j = 0$ for all $j \neq j^*$.

Now, the imputed values are given by

$$b_i^{imp} = \sum_{j=1}^{k} w_j \boldsymbol{z}_j, \quad i = 1, \ldots, m.$$

Naturally, in all the cases, we can skip the procedure, if $(\boldsymbol{a}_i, b_i)$ has no missing value in b_i and repeat the procedure (with different $(\boldsymbol{a}_i, b_i)$) if object $\hat{\boldsymbol{a}}_i$ has more than one missing value.

While the first approach is the most simple to compute it suffers the same drawback than just a single regression approach: all the missing values of a single feature are imputed to one regression line. The second and third approaches make it possible to better utilize the cluster structure obtained. The problem with these approaches and

large data sets is the computational burden when computing nearest neighbours. In order to make the implementation more efficient we, instead of using every data point, select the maximum number of points $l_{max} \leq m$ that we are randomly looking trough when seeking for nearest neighbours.

4 Numerical Experiments

The proposed algorithm IVIACLR was tested using some artificial and real world data sets. The IVIACLR -algorithm is compared to some commonly used methods for imputation: the mean imputation, regression imputation, and MICE [2, 41, 53].

The IVIACLR is implemented in Fortran 95 and compiled using `gfortran`, the GNU Fortran compiler. The mean and regression imputations are obtained as initial imputations for the IVIACLR (i.e. we select the initial imputation to be either the mean or the regression and set the number of regression functions to zero in the actual CLR procedure). For MICE the build-in R-implementation with default parameters is used [49].

4.1 Data Sets

To test and compare the above mentioned imputation methods we have used three artificial and five real life data sets. We generated incomplete data sets with varying percentages of missing values (from 5% to 45%) by randomly removing some of the values from original complete data sets. Nevertheless, all data points need to have at least one reference attribute in it. For all original data sets we performed 10 runs with all percentages of missing values. That is, in 10 runs the original complete data and the percentage of missing values are the same but different values are missing. The results given are averaged over these 10 runs.

Artificial Data Set. To see the performance of the new imputation method in different types of data, the synthetic data sets were generated to be very different from each other. The first data set D500 has no structure in it. It is generated using uniform distribution with the mean value 0 and standard deviation 1. The number of data points is 500 and the number of features is 4. In addition, we used five regression functions when testing the IVIACLR. In turn, the second synthetic data set U500 has three clearly separated clusters (see, Figure 7). It contains 500 data points and 2 features. For this data set we tested the IVIACLR with two and three regression functions. The third data set U2500 contains 2500 data points and 20 features. It has 5 clusters, some of witch slightly overlaps each other (nevertheless, the structure is still clear). Naturally, five regression functions were used when testing the IVIACLR.

Real World Data Sets. We use one small and four larger real world data sets in our experiments. Their names, numbers of data points m and features n, and the optimal

number of clusters k in the data sets are given in Table 1. All these data sets can be found from [31].

Table 1: Real world data sets.

Data	m	n	k
Iris	150	4	3
TSPLIB1060	1060	2	5
Red wine quality	1599	11	6[1]
Abalone	4177	8	2[2]
White wine quality	4898	11	7[1]

[1] Number of clusters in [28]. [2] Number of clusters in [9].

4.2 Evaluation criteria

Imputation methods were compared using four evaluation criteria:

1. *Root mean square error* (RMSE) measures the difference between true and imputed values. It is computed by the formula

$$\text{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{a}_i^{obs} - \boldsymbol{a}_i^{imp})^2},$$

   where $\boldsymbol{a}_i^{obs}$ and $\boldsymbol{a}_i^{imp}$ are the observed and imputed values, respectively, and $a_{ij}^{imp} = a_{ij}^{obs}$, $j \in \{1, 2, \ldots, n\}$, if the value a_{ij} is not missing.

2. *Mean absolute error* (MAE) measures the average magnitude of the errors and it is computed by the formula

$$\text{MAE} = \frac{1}{m}\sum_{i=1}^{m}|\boldsymbol{a}_i^{obs} - \boldsymbol{a}_i^{imp}|.$$

3. *Unsupervised classification error* (UCE) assesses the preservation of an internal structure. That is, the UCE measures how well the clustering of the complete data set is preserved when clustering the imputed data set. We define the UCE as

$$\text{UCE} = \ \% \text{ of misclassified samples.}$$

   Here we use the LMBM-CLUST [29] as a clustering method (The LMBM-CLUST is available at `http://napsu.karmitsa.fi/clustering/`).

4. *Cluster center misplacement* (CCM) measures the distance between the centers of clusters in the complete data set and in the imputed one. We define the CCM as

$$\text{CCM} = \frac{1}{k}\sum_{i=1}^{k}|\boldsymbol{c}_i^{orig} - \boldsymbol{c}_i^{imp}|,$$

where $\boldsymbol{c}_i^{orig}$ and $\boldsymbol{c}_i^{imp}$ are the centers of ith clusters in original and imputed data sets, respectively, and k is the number of clusters.

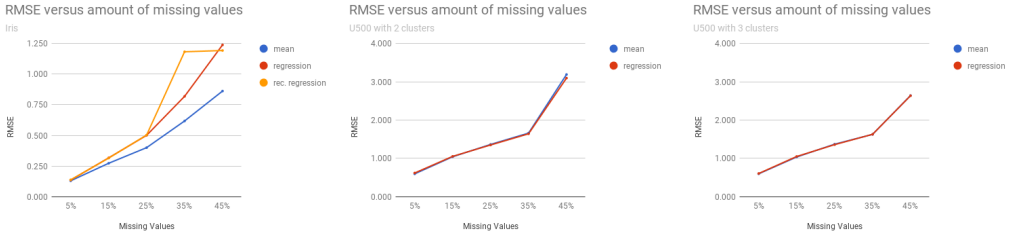
Note that the sole CCM does not give much information about the accuracy of the imputation. Nevertheless, it supports and reinforces the UCE by telling if also the centers of clusters are preserved. That is, if we obtain both a small UCE and a small CCM the imputation can be considered accurate. Otherwise, a small UCE but large CCM means that there is some bias in imputed values, a large UCE with small CCM indicates that the overall structure of the data set is preserved but some imputed values are incorrect, and both the UCE and CCM large indicates that the structure of data set is lost.

4.3 Results

The tables of results can be found in Appendix. Here we visualise the most relevant ones in Figures 2–21 and draw some conclusions.

Parameters for IviaCLR. We start our experiments by searching a good combination of parameters for the IVIACLR: i.e. the type of an initial imputation, the number of outerloops o_{max} in Algorithm 1, the type of a prediction method, and the number of nearest neighbours in the prediction phase. As noted in subsection 3.3, the first prediction method introduced suffers from the drawback that all missing values of a single feature are imputed to a sole regression line similarly to the regression imputation (see Figures 8(d) and 9(d) for illustration of the regression imputation). Our pre-preliminary tests confirmed that this prediction method does not work properly in missing value imputation. Thus, we omit it from our testing. In addition, in the first version of the code we do not have the tolerance ε for the change (see Algorithm 1) and we use $l_{max} = 150$ (see subsection 3.3) with all data sets. In order to find a good combination of parameters we use data sets Iris and U500, and compare results in light of RMSEs only.

Figure 2 illustrates the RMSE with increasing amount of missing data for different initial imputations. From these results we see that in Iris data set the $mean$ works clearly best as an initial imputation. In U500 data set, which has only two features, the *regression* and *recursive regression* give the same initial imputations, and the results show no clear preference to either the *mean* or *regression* as the initial imputation. In Figure 2 we have used the *RMSE based local weighting* with $knn = 5$ as prediction



(a) Iris (b) U500 with 2 cluster (c) U500 with 3 clusters

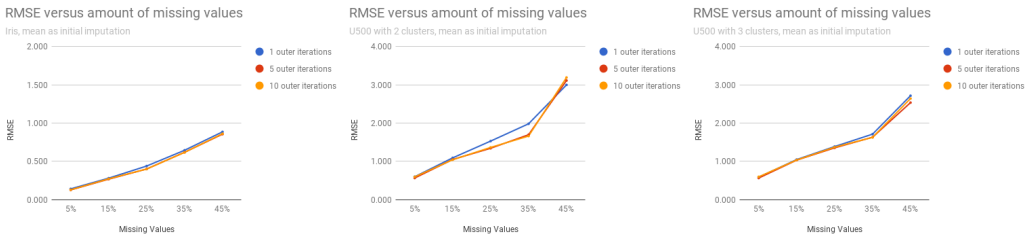
Figure 2: IVIACLR: RMSE for different initial imputations.

and $o_{max} = 10$. The other parameter combinations reveal the same trends but smaller o_{max} favor *mean* also in U500 data set (see Figures 3 and 4 and Appendix).

Next we study the effect of outer iterations. The results are given in Figures 3 and 4 with the *mean* and *regression* as initial imputations, respectively. In addition, we have used the *RMSE based local weighting* with $knn = 5$. For the results with other parameters, see Appendix. We see that $o_{max} = 1$ is not enough even with the data set U500 with only two features. In addition, the larger o_{max} may prevent a "bad" initial solution. In what follows, we use $o_{max} = 10$ for small data sets and, since this choice naturally means more computational burden, $o_{max} = 5$ is used for larger data sets.

There is no big difference in results with different prediction methods as can be seen from Figure 5. In Iris and U500, when 3 regression functions are used with the IVIACLR, the *RMSE based local weighting* seems to be slightly better as prediction. Thus, for the rest of the tests we use the *RMSE based local weighting*.

Figure 6 illustrates the RMSE with increasing amount of missing data for different values of knn. From the figure we see that for up to 35% of data missing $knn = 5$ is the best choice. However, with very large parts of data missing the usage of smaller knn might give more accurate imputations. The other parameters used here are those obtained above. That is, the *mean* as initial imputation, *RMSE based local weighting* as prediction, and $o_{max} = 10$. The other parameter combinations reveal the same trends (see Appendix). Therefore, for the rest of the tests we use the value $knn = 5$.



(a) Iris (b) U500 with 2 cluster (c) U500 with 3 clusters

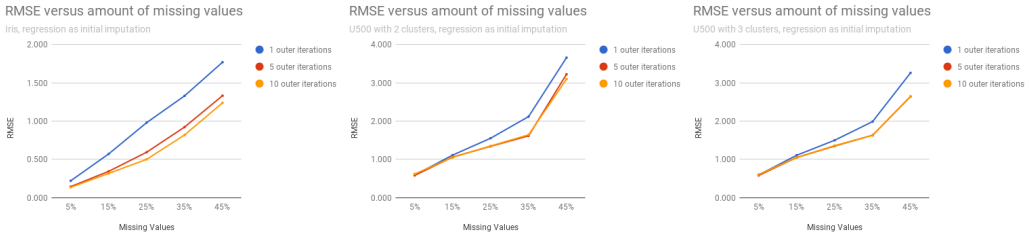
Figure 3: IVIACLR: RMSE for different numbers of outer iterations, *mean* as an initial imputation.

From Figures 2–6 (b) and (c) we see that for small amount of missing data 2 regression functions is enough in U500. However, with 45% 3 regression functions give clearly more accurate imputation.

Different types of data. Next we study the behavior of the IVIACLR and the other imputation methods in different types of data. For this purposis we use the three artificial data sets D500, U500, and U2500. The original data set U500 as well as the imputed data sets with 5% and 45% of missing values are illustrated in Figures 7 – 9, respectively. The RMSE, MAE, UCE, and CCM of different imputation methods are illustrated in Figure 10 and the more comprehensive tables of the results are given in Appendix. We recall that the RMSE, MAE, UCE, and CCM are averaged over 10 runs with different values missing in the data sets. For the imputed data sets in figures we just selected the first data set in our collection with 5% or 45% of missing values (i.e. with all algorithms the data sets with same values missing are used in figures). From these figures we can clearly see the superiority of the IVIACLR when the data is clearly structured. With 5% of missing values it clearly misplaces only one value (see Figure 8(a)). This misplacement is due to prediction where some of the knn neighbours belong to the different cluster. In addition, the structure of the data set can still be seen in the imputed data set even when almost half of the data is missing (see Figure 9(a)). MICE works quite well with 5% of missing data. However, it misplaces the same data point than the IVIACLR and the absolute error is greater. From Figure 10 we see that the IVIACLR always has the smallest RMSE and the MAE is similar to that of MICE. In addition, the UCE and CCM show that the IVIACLR is clealy the best in preserving the original structure of the data set up to 35% of missing values: less than 5% of data points are clustered to some other cluster than with the complete data set and cluster centers are approximately the same.

Figure 11 shows the RMSE, MAE, UCE, and CCM in data set D500 with no structure. From this figure we conclude the obvious result that if no clusters or other structure is present then the mean and regression imputations are as good as any other method. In fact, our results show a little advantage to the mean and regression over the more sophisticated methods MICE and IVIACLR. Although, we compute more regression functions in the IVIACLR, and thus cover the space more densely, the fact that points are spread out randomly means that the prediction phase of the IVIACLR fails.

Figures 12 and 13 show the results in U2500 data set with larger numbers of data points and features. Figure 12 gives results with all imputation methods while Figure 13 compares only MICE and the IVIACLR. Similarly to U500 data set, the IVIACLR produces most accurate imputation due to clear structure of the data set. Note that Figures 12(c) and 12(d) show us the usefulness of the CCM criterion: sole UCE would indicate as good imputation with mean as with MICE and IVIACLR up to 25% of missing values but the CMM indicates large bias in values imputed by the mean.

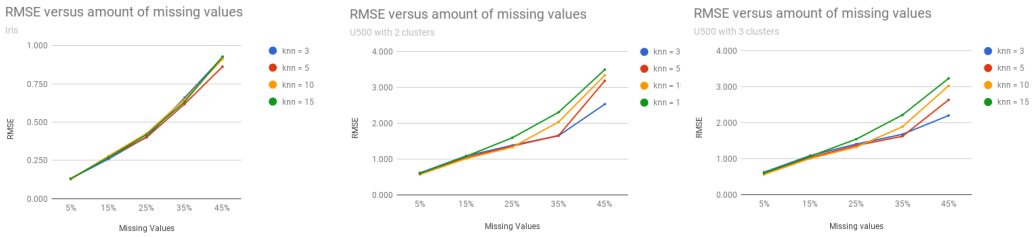


(a) Iris (b) U500 with 2 cluster (c) U500 with 3 clusters

Figure 4: IVIACLR: RMSE for different numbers of outer iterations, *regression* as an initial imputation.

(a) Iris (b) U500 with 2 cluster (c) U500 with 3 clusters

Figure 5: IVIACLR: RMSE with different predictions.



(a) Iris (b) U500 with 2 cluster (c) U500 with 3 clusters

Figure 6: IVIACLR: RMSE with for different values of knn.

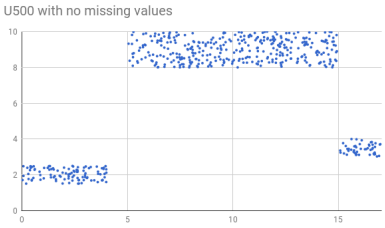
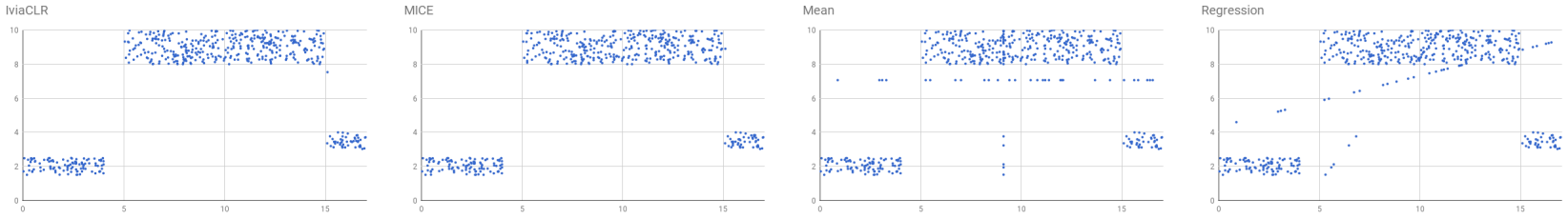
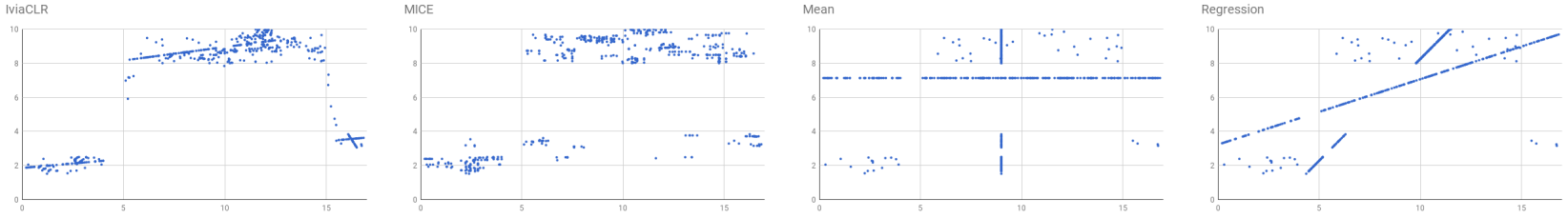


Figure 7: Original U500



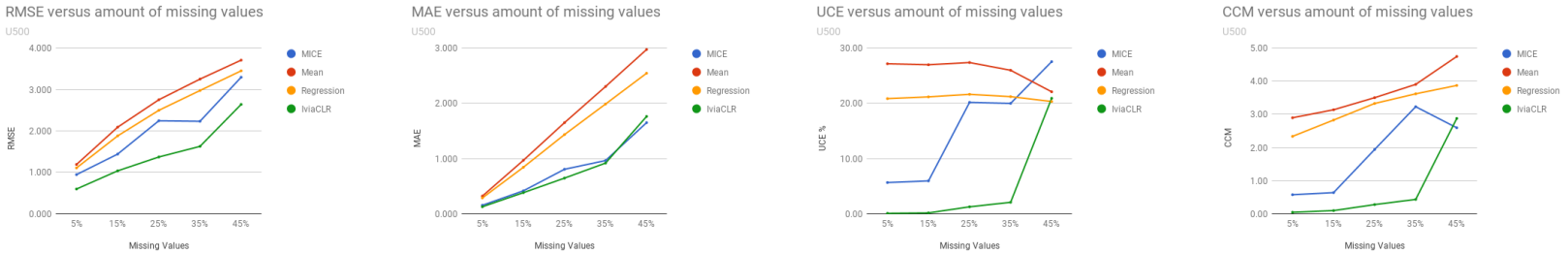
(a) Imputation by IVIACLR (b) Imputation by MICE (c) Imputation by Mean (d) Imputation by Regression

Figure 8: U500: imputed data sets with 5% of missing data.



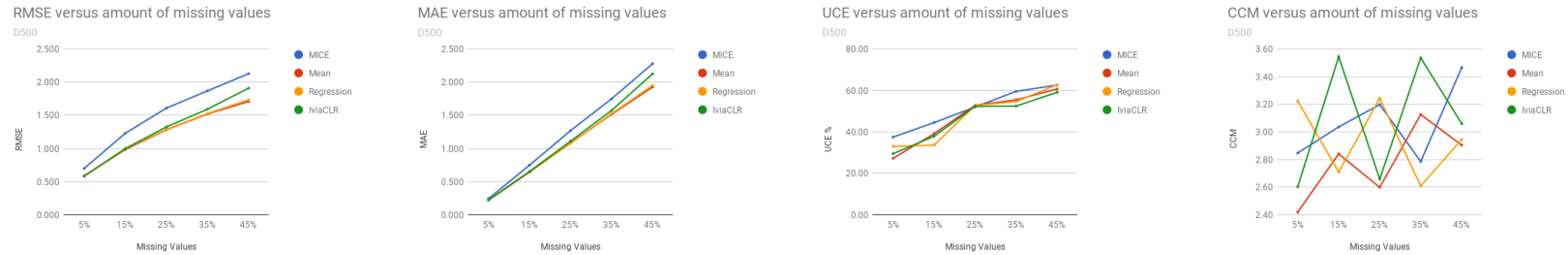
(a) Imputation by IVIACLR (b) Imputation by MICE (c) Imputation by Mean (d) Imputation by Regression

Figure 9: U500: imputed data sets with 45% of missing data.



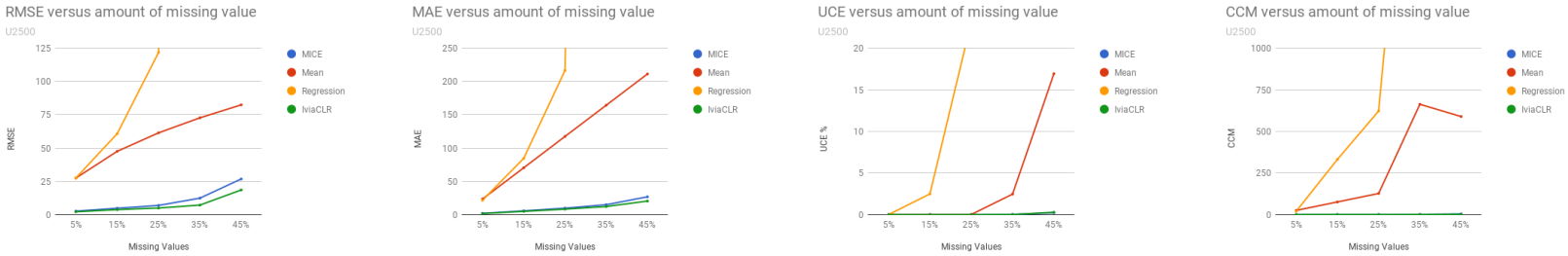
(a) RMSE (b) MAE (c) UCE (d) CCM

Figure 10: U500: RMSE, MAE, UCE, and CCM versus the number of missing values.



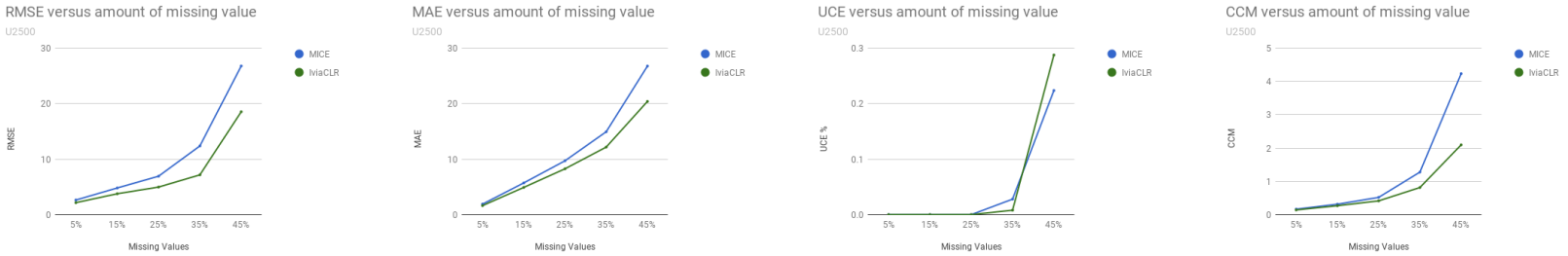
(a) RMSE (b) MAE (c) UCE (d) CCM

Figure 11: D500: RMSE, MAE, UCE, and CCM versus the number of missing values.



(a) RMSE (b) MAE (c) UCE (d) CCM

Figure 12: U2500: RMSE, MAE, UCE, and CCM versus the number of missing values.



(a) RMSE (b) MAE (c) UCE (d) CCM

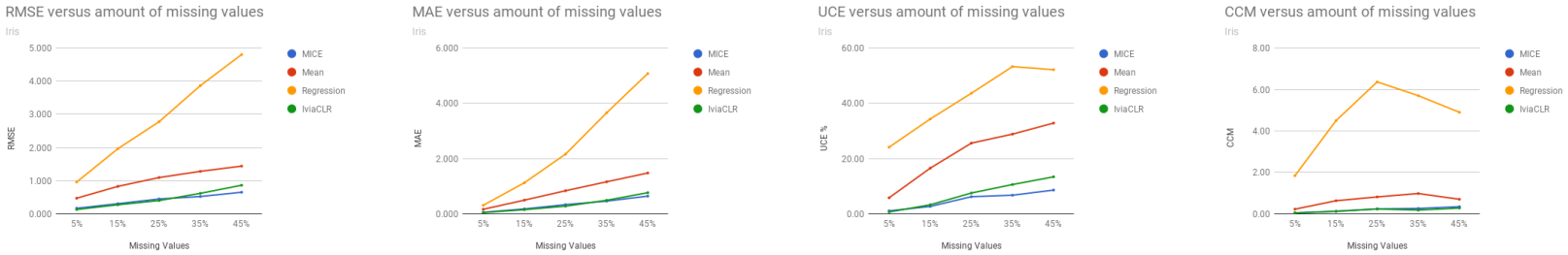
Figure 13: U2500: RMSE, MAE, UCE, and CCM versus the number of missing values for MICE and IVIACLR.

Real world data. In addition to artificial data sets with known structures, we compare IVIACLR to the other imputation methods using some real world data sets. Figures 14 and 18–21 show the RMSE, MAE, UCE, and CCM in Iris, TSPLIB1060, Red Wine, Abalone, and White Wine data sets, respectively. In addition, Figures 15 – 17 illustrate the original data set TSPLIB1060 and the imputed data sets with 5% and 45% of missing data.

With Iris data the proposed method IVIACLR gives the smallest errors with less than 35% of missing values. With 35% and 45% MICE gives smaller errors. Nevertheless, the differences between the IVIACLR and MICE are small in both directions. In addition, the UCE and CCM show that the IVIACLR and MICE work similar with less or equal to 25% of missing values misclassifying less than 10% of data points. With larger percentages of missing values MICE produces more accurate imputations. In addition, both the IVIACLR and MICE produce clearly more accurate imputations than the mean and regression.

In TSPLIB1060 the largest errors are always obtained with MICE. Although the UCE and CCM indicate that MICE preserves the original cluster structure more accurately than the other methods, the percentages of misclassified data points is huge ($> 40\%$) when there are more than 25% of missing values. This is true for all the tested methods. With less than 25% of missing values the IVIACLR works as good as MICE in terms of the UCE and CCM, and it always has smaller errors. In addition, the mean and regression imputations produce relatively small errors in TSPLIB1060 data set. This is due to the large cluster of data points at the middle of the data space (see Figure 15) which makes mean an average good approximation for missing values. Nevertheless, the mean and regression imputations are even worse in preserving the cluster structure than the other two imputation methods. Especially, with greater percentages of missing data, the CCM shows very large bias in centers of clusters with the mean and regression imputations.

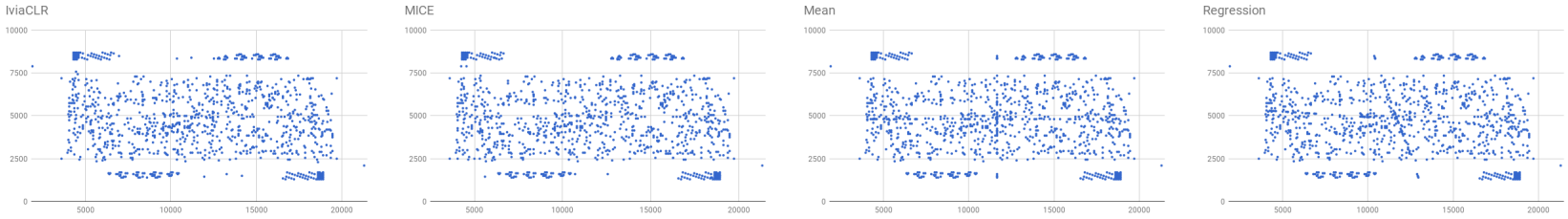
For the rest of the data sets the results obtained with the regression imputation are very large compared to others (see Appendix). To make the figures more illustrative, we have omitted these results. In Red wine quality data set with less than 25% of missing values the proposed method IVIACLR is the best imputation method according to all measured evaluation criteria. MICE has the largest errors but with large percentages of missing data it preserves the original structure of the data set most accurately yet, again, the percentages of misclassified data points are large with all the methods with more than 25% of missing values. In Abalone the IVIACLR again has the smallest errors, but the UCE and CCM indicate that the IVIACLR does not preserve the original structure of the data set as well as the other methods (but regression). However, the percentage of misclassified data points here is smaller than that in TSPLIB1060 or Red wine quality.



(a) RMSE (b) MAE (c) UCE (d) CCM

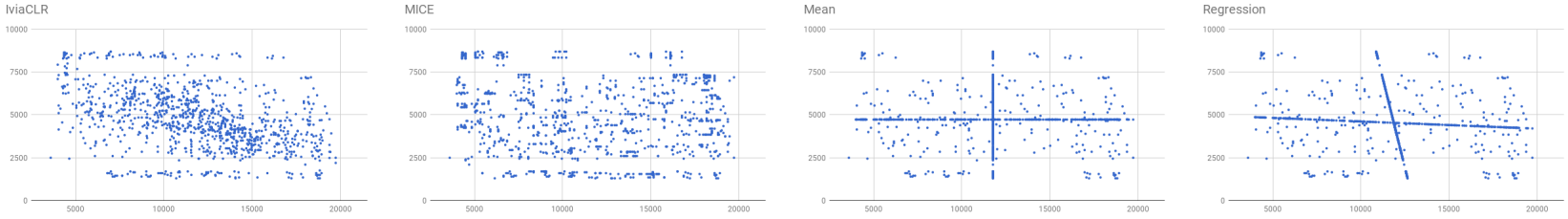
Figure 14: Iris: RMSE, MAE, UCE, and CCM versus the number of missing values.

Figure 15: Original TSPLIB1060



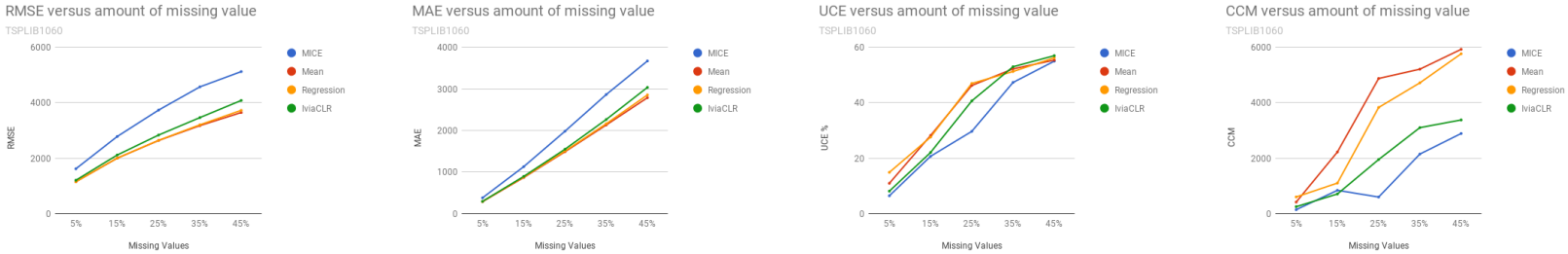
(a) Imputation by IVIACLR (b) Imputation by MICE (c) Imputation by Mean (d) Imputation by Regression

Figure 16: TSPLIB1060: imputed data sets with 5% of missing data.



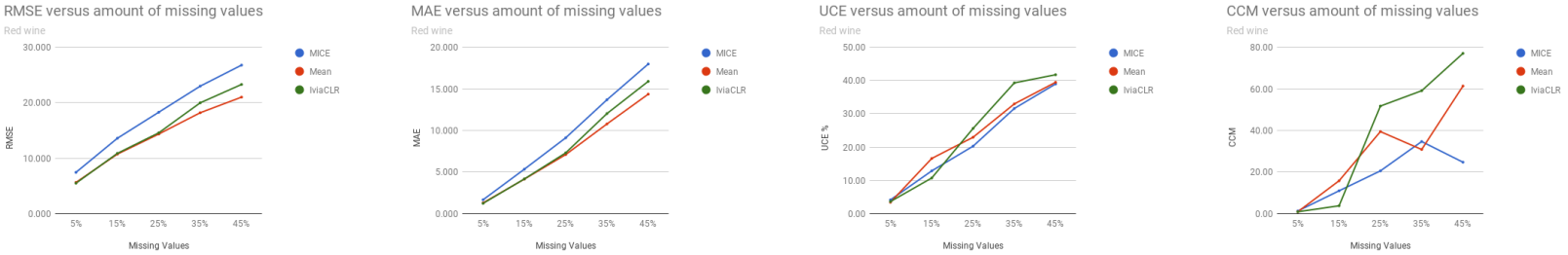
(a) Imputation by IVIACLR (b) Imputation by MICE (c) Imputation by Mean (d) Imputation by Regression

Figure 17: TSPLIB1060: imputed data sets with 45% of missing data.



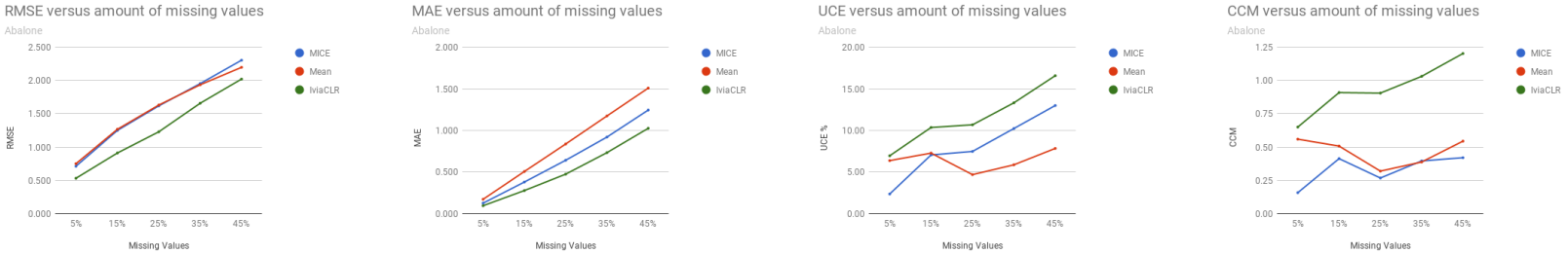
(a) RMSE (b) MAE (c) UCE (d) CCM

Figure 18: TSPLIB1060: RMSE, MAE, UCE, and CCM versus the number of missing values.



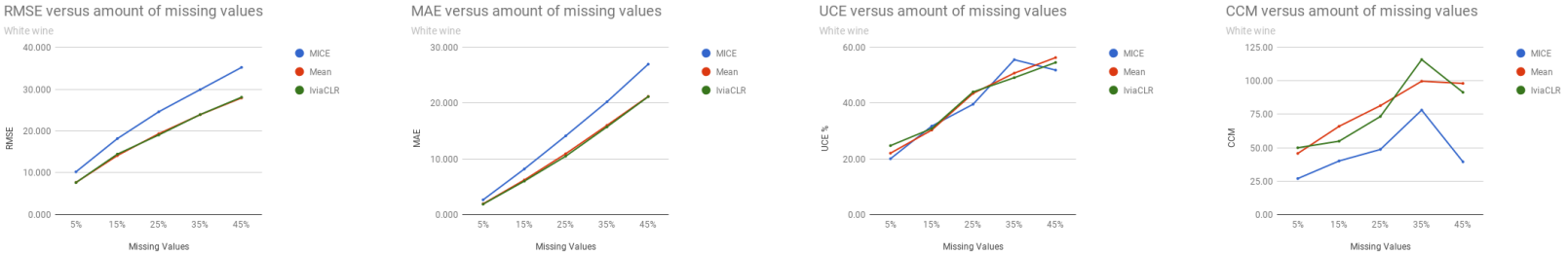
(a) RMSE (b) MAE (c) UCE (d) CCM

Figure 19: Red Wine: RMSE, MAE, UCE, and CCM versus the number of missing values.



(a) RMSE (b) MAE (c) UCE (d) CCM

Figure 20: Abalone: RMSE, MAE, UCE, and CCM versus the number of missing values.



(a) RMSE (b) MAE (c) UCE (d) CCM

Figure 21: White Wine: RMSE, MAE, UCE, and CCM versus the number of missing values.

In Abalone the mean preserves the structure of the original data set most accurately, especially, with large percentages of data missing. In White wine quality MICE has again greatest error. The UCEs of the methods (but the regression) are similar but the CCM indicates less bias with MICE than with the other methods.

We conclude that the proposed method IVIACLR worked best with small and moderate amount of missing data being usually the most accurate imputation method tested. More so, if there is a clear structure in the data set in question. With very large parts of data missing MICE usually give more accurate imputations at least in terms of the UCE and CCM.

5 Conclusions and Discussion

A new approach IVIACLR (imputation via clusterwise linear regression) for imputing missing features of incomplete data was proposed in this paper. The approach is based on clusterwise linear regression and it simultaneously finds optimal clusters within the data and their associated regression functions. The idea is to approximate missing values using only those data points that are somewhat similar to the incomplete data object. In addition, we introduced a new cluster center misplacement (CCM) criterion that can be used together with the well-known unsupervised classification error (UCE) to measure the bias in the imputed values.

The IVIACLR was tested and compared to other imputations methods using the root mean square error (RMSE), mean absolute error (MAE), and the above mentioned UCE and CCM. The results confirm that the IVIACLR usually finds smaller errors (RMSE and MAE) than the well-known imputation method MICE. In addition, with small and moderate percentages of missing values (say $\leq$25%) the UCE and CCM indicate that the original structure of data set imputed with the IVIACLR is well preserved. With larger percentages of missing data MICE usually produces more accurate imputation in terms of the UCE and CCM but not necessary the RMSE and MAE. We conclude that the proposed algorithm IVIACLR produces the most accurate imputations in data sets with clear structure and small or moderate amount of missing values.

In the current version of IVIACLR we only use linear regression and consider continuous numeric data. Nevertheless, it would be possible to generalize our approach to deal with different data types (e.g. binary) if some other regression model was used. In addition, although now used as a single imputation method, the proposed method could be used as multiple imputation (MI) method (comparably e.g. with MICE) by taking into account all the intermediate results obtained during the clusterwise linear regression process, by taking predictions provided by different regression functions as multiple imputations, using different prediction methods, and/or using a different initial imputation.

Acknowledgments

The work was financially supported by the Academy of Finland (Project No. 289500, 294002, and 313266) and Australian Research Counsil's Discovery Projects funding scheme (Project No. DP140103213).

References

[1] ANDRIDGE, R. R., AND LITTLE, R. J. A. A review of hot deck imputation for survey non-response. *International Statistical Review 78*, 1 (2010), 40–64.

[2] AZUR, M., STUART, E., FRANGAKIS, C., AND LEAF, P. Multiple imputation by chained equations: What is it and how does it work? *International Journal of Methods in Psychiatric Research 20*, 1 (2011), 40–49.

[3] BAGIROV, A. M., KARMITSA, N., AND MÄKELÄ, M. M. *Introduction to Nonsmooth Optimization: Theory, Practice and Software*. Springer, 2014.

[4] BAGIROV, A. M., MAHMOOD, A., AND BARTON, A. Prediction of monthly rainfall in victoria, australia: Clusterwise linear regression approach. *Atmospheric Research 188* (2017), 20–29.

[5] BAGIROV, A. M., UGON, J., AND MIRZAYEVA, H. Nonsmooth nonconvex optimization approach to clusterwise linear regression problems. *European Journal of Operational Research 229*, 1 (2013), 132–142.

[6] BAGIROV, A. M., UGON, J., AND MIRZAYEVA, H. An algorithm for clusterwise linear regression based on smoothing techniques. *Optimization Letters 9*, 2 (2015), 375–390.

[7] BAGIROV, A. M., UGON, J., AND MIRZAYEVA, H. G. Nonsmooth optimization algorithm for solving clusterwise linear regression problem. *Journal of Optimization Theory and Applications 164* (2015), 755–780.

[8] BATISTA, G., AND MONARD, M. C. A study of k-nearest neighbor as an imputation method. In *Hybrid Intelligent Systems*, Abraham, A. et. al. , Ed. IOS Press, 2002, pp. 251–260.

[9] C ZHANG, C., QIN, Y., ZHU, X., ZHANG, J., AND S. Z. Clustering-based missing value imputation for data preprocessing. Industrial Informatics, 2006 IEEE International Conference, 2006.

[10] CARBONNEAU, R., CAPOROSSI, G., AND HANSEN, P. Globally optimal clusterwise regression by mixed logical-quadratic programming. *European Journal of Operational Research 212* (2011), 213–222.

[11] CARBONNEAU, R., CAPOROSSI, G., AND HANSEN, P. Extensions to the repetitive branch-and-bound algorithm for globally-optimal clusterwise regression. *Computers and Operations Research 39*, 11 (2012), 2748–2762.

[12] CHI, J. T., CHI, E. C., AND BARANIUK, R. G. k-POD: A method for k-means clustering of missing data. *The American Statistician 70* (2016), 91–99.

[13] CLARKE, F. H. *Optimization and Nonsmooth Analysis*. Wiley-Interscience, New York, 1983.

[14] DESARBO, W., AND CRON, W. A maximum likelihood methodology for clusterwise linear regression. *Journal of Classification 5*, 2 (1988), 249–282.

[15] DESARBO, W., OLIVER, R., AND RANGASWAMY, A. A simulated annealing methodology for clusterwise linear regression. *Psychometrika 54*, 4 (1989), 707–736.

[16] ENDERS, C. *Applied Missing Data Analysis*. The Guilford Press, 2010.

[17] GABRYS, B. Neuro-fuzzy approach to processing inputs with missing values in pattern recognition problems. *International Journal of Approximate Reasoning 30*, 3 (2002), 149–179.

[18] GAFFNEY, S., AND SMYTH, P. Trajectory clustering using mixtures of regression models. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining* (1999), S. Chaudhuri and D. Madigan, Eds., New York, pp. 63–72.

[19] GAO, H., LIU, X.-W., PENG, Y.-X., AND JIAN, S.-L. Sample-based extreme learning machine with missing data. *Mathematical Problems in Engineering 2015* (2015), 1–11.

[20] GARCÌA-ESCUDERO, L., GORDALIZA, A., MAYO-ISCAR, A., AND SAN MARTIN, R. Robust clusterwise linear regression through trimming. *Computational Statistics and Data Analysis 54* (2010), 3057–3069.

[21] GUPTA, A., AND LAM, M. S. Estimating missing values using neural networks. *Journal of the Operational Research Society 47*, 2 (1996), 229–238.

[22] HAARALA, M., MIETTINEN, K., AND MÄKELÄ, M. M. New limited memory bundle method for large-scale nonsmooth optimization. *Optimization Methods and Software 19*, 6 (2004), 673–692.

[23] HAARALA, N., MIETTINEN, K., AND MÄKELÄ, M. M. Globally convergent limited memory bundle method for large-scale nonsmooth optimization. *Mathematical Programming 109*, 1 (2007), 181–205.

[24] HATHAWAY, R., AND BEZDEK, J. Fuzzy c-means clustering of incomplete data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 31*, 1 (2001), 735 – 744.

[25] HAZAN, E., LIVNI, R., AND MANSOUR, Y. Classification with low rank and missing data. In *Proceedings of the 32nd International Conference on Machine Learning* (Lille, France, 2015).

[26] HE, L., HUANG, G. H., AND LU, H. W. Health-risk-based groundwater remediation system optimization through clusterwise linear regression. *Environmental Science & Technology 42*, 24 (2008), 9237–9243.

[27] HEJAZI, M., AL-HADDAD, S. A. R., SINGH, Y. P., AND AZIZ, A. F. Multi-class support vector machines for classification of ECG data with missing values. *Applied Artificial Intelligence 29*, 7 (2015), 660–674.

[28] KARMITSA, N., BAGIROV, A. M., AND TAHERI, S. Limited memory bundle method for solving large clusterwise linear regression problems. TUCS Technical Report, No. 1172, Turku Centre for Computer Science, Turku, 2016. The report is available online at http://tucs.fi/publications/view/?pub_id=tKaBaTa16c.

[29] KARMITSA, N., BAGIROV, A. M., AND TAHERI, S. Mssc clustering of large data using the limited memory bundle method. TUCS Technical Report, No. 1164, Turku Centre for Computer Science, Turku, 2016. The report is available online at http://tucs.fi/publications/view/?pub_id=tKaBaTa16b.

[30] LI, D., DEOGUN, J., SPAULDING, W., AND SHUART, B. Towards missing data imputation: A study of fuzzy k-means clustering method. In *Rough Sets and Current Trends in Computing. 4th International Conference, RSCTC 2004, Uppsala, Sweden.*, S. Tsumoto, R. Słowiński, J. Komorowski, and J. Grzymała-Busse, Eds. Springer, 2004, pp. 573–579.

[31] LICHMAN, M. UCI machine learning repository. Available in web page <URL: `http://archive.ics.uci.edu/ml`>, University of California, Irvine, School of Information and Computer Sciences, 2013. (April 8th, 2016).

[32] LITTLE, R., AND RUBIN, D. *Statistical Analysis with Missing Data*, 2nd edition ed. John Wiley and Sons, 2002.

[33] MOJIRSHEIBANI, M., AND REESE, T. Kernel regression estimation for incomplete data with applications. *Statistical Papers 58*, 1 (2015), 185–209.

[34] ORDIN, B., AND BAGIROV, A. A heuristic algorithm for solving the minimum sum-of-squares clustering problems. *Journal of Global Optimization 61*, 2 (2015), 341–361.

[35] PARK, Y., JIANG, Y., KLABJAN, D., AND WILLIAMS, L. Algorithms for generalized cluster-wise linear regression. *ArXiv e-prints* (2016).

[36] PATIL, B. M., JOSHI, R. C., AND TOSHNIWAL, D. Missing value imputation based on k-mean clustering with weighted distance. In *Contemporary Computing. IC3 2010. Communications in Computer and Information Science, vol 94*, R. S. et al., Ed. Springer, Berlin, Heidelberg, 2010, pp. 600–609.

[37] PELCKMANS, K., BRABANTER, J. D., SUYKENS, J. A. K., AND MOOR, B. D. Handling missing values in support vector machines classifiers. *Neural networks 18*, 5–6 (2005), 684–692.

[38] POGGI, J.-M., AND PORTIER, B. PM10 forecasting using clusterwise regression. *Atmospheric Environment 45*, 38 (2011), 7005–7014.

[39] PREDA, C., AND SAPORTA, G. Clusterwise pls regression on a stochastic process. *Computational Statistics & Data Analysis 49* (2005), 99–108.

[40] QIN, Y., ZHANG, S., ZHU, X., ZHANG, J., AND ZHANG, C. Semi-parametric optimization for missing data imputation. *Applied Intelligence 27* (2007), 79–88.

[41] RAGHUNATHAN, T., LEPKOWSKI, J., VAN HOEWYK, J., AND SOLENBERGER, P. A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey Methodology 27*, 1 (2001), 85–95.

[42] SHARPE, P. K., AND SOLLY, R. J. Dealing with missing values in neural network-based diagnostic systems. *Neural Computing and Applications 3*, 2 (1995), 73–77.

[43] SHIN-MU TSENG, KUO-HO WANG, AND CHIEN-I. LEE. A pre-processing method to deal with missing values by integrating clustering and regression techniques. *Applied Artificial Intelligence 17*, 5–6 (2003), 535–544.

[44] SILVA-RAMÍREZA, E.-L., PINO-MEJÍASB, R., LÓPEZ-COELLOA, M., AND CUBILES-DE-LA VEGAC, M.-D. Missing value imputation on missing completely at random data using multilayer perceptrons. *Neural Networks 24*, 1 (2011), 121–129.

[45] SPÄTH, H. Algorithm 39: Clusterwise linear regression. *Computing 22* (1979), 367–373.

[46] SPÄTH, H. Algorithm 48: A fast algorithm for clusterwise linear regression. *Computing 29* (1981), 175–181.

[47] TANG, W., HE, H., AND GUNZLER, D. Kernel smoothing density estimation when group membership is subject to missing. *Journal of Statistical Planning and Inference 142*, 3 (2012), 685–694.

[48] TUTZ, G., AND RAMZAN, S. Improved methods for the imputation of missing data by nearest neighbor methods. *Computational Statistics and Data Analysis 90* (2015), 84–99.

[49] VAN BUUREN, S. MICE. Available in web page <URL: https://cran.r-project.org/web/packages/mice/mice.pdf> (February 28th, 2018).

[50] WAGSTAFF, K. Clustering with missing values: No imputation required. In *Classification, Clustering, and Data Mining Applications. Proceedings of the Meeting of the International Federation of Classification Societies (IFCS), Illinois Institute of Technology, Chicago*, B. D. M. F.R. A. P. and G. W. Eds. Springer, 2004, pp. 649–658.

[51] WANG, Q. H., AND RAO, R. Empirical likelihood-based inference under imputation for missing response data. *Annals of Statistics 30* (2002), 896–924.

[52] WEDEL, M., AND KISTEMAKER, C. Consumer benefit segmentation using clusterwise linear regression. *International Journal of Research in Marketing 6*, 1 (1989), 45–59.

[53] WHITE, I., ROYSTON, P., AND WOOD, A. Multiple imputation using chained equations: Issues and guidance for practice. *Statistics in Medicine 30*, 4 (2011), 377–399.

[54] WILLIAMS, D., LIAO, X., XUE, Y., CARIN, L., AND KRISHNAPURAM, B. On classification with incomplete data. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 3 (2007), 427–436.

[55] WILLIAMS, R. Missing data part 1: Overview, traditional methods. University of Notre Dame, available in web page <URL: http://www3.nd.edu/∼rwilliam/stats2/l12.pdf>, 2015. (February 14th, 2017).

[56] YAN, Y., ZHANG, Y., CHEN, J., AND ZHANG, Y. Incomplete data classification with voting based extreme learning machine. *Journal Neurocomputing 193*, C (2016), 167–175.

[57] YAO, L., AND WENG, K. Imputation of incomplete data using adaptive ellipsoids with linear regression. *Journal of Intelligent & Fuzzy Systems 29* (2015), 253–265.

[58] ZHANG, L., LU, W., LIU, X., PEDRYCZ, W., AND ZHONG, C. Fuzzy c-means clustering of incomplete data based on probabilistic information granules of missing values. *Knowledge-Based Systems 99*, C (2016), 51–70.

Appendix

We give here the results obtained in our numerical experiments. Unless said otherwise we have used default parameters for IVIACLR. That is, *mean* as initial imputation, *RMSE based local weighting* as prediction, $knn = 5.$ and $o_{max} = 10$ (for small data sets) or $o_{max} =$ (for large data sets). For those results where also other parameter than the default ones are used we have used the following coding:

$$i_{imp} - i_{pred} - knn - o_{max},$$

where i_{imp} is the type or initial imputation:

- $i_{imp} = 0$: mean;
- $i_{imp} = 1$: regression;
- $i_{imp} = 2$: regular regression;

i_{pred} is the prediction method:

- $i_{pred} = 0$: Local weighting;
- $i_{pred} = 1$: RMSE based local weighting;

and knn and o_max are the number of nearest neighbours and the number of outer iterations, respectively. For example "0-1-05-10" means that we have used *mean* as initial imputation, *RMSE based local weighting* as prediction, $knn = 5.$ and $o_{max} = 10$.

Table 2: IVIACLR: RMSE with increasing knn.

Iris					
knn	5%	15%	25%	35%	45%
3	0.131	0.260	0.403	0.659	0.923
5	0.130	0.273	0.400	0.618	0.862
10	0.126	0.279	0.426	0.650	0.912
15	0.131	0.271	0.418	0.632	0.928

U500 with 2 clusters					
knn	5%	15%	25%	35%	45%
3	0.616	1.090	1.387	1.656	2.538
5	0.597	1.044	1.366	1.662	3.191
10	0.569	1.016	1.336	2.039	3.346
15	0.601	1.079	1.599	2.310	3.504

U500 with 3 clusters					
knn	5%	15%	25%	35%	45%
3	0.627	1.090	1.412	1.683	2.207
5	0.598	1.037	1.372	1.628	2.642
10	0.566	1.010	1.328	1.894	3.034
15	0.606	1.069	1.549	2.222	3.239

Table 3: IVIACLR: MAE with increasing knn.

Iris					
knn	5%	15%	25%	35%	45%
3	0.044	0.147	0.279	0.508	0.792
5	0.044	0.150	0.277	0.489	0.766
10	0.043	0.155	0.292	0.510	0.799
15	0.044	0.150	0.288	0.507	0.817

U500 with 2 clusters					
knn	5%	15%	25%	35%	45%
3	0.125	0.394	0.645	0.910	1.630
5	0.125	0.383	0.648	0.951	2.169
10	0.123	0.380	0.666	1.248	2.233
15	0.135	0.422	0.821	1.404	2.283

U500 with 3 clusters					
knn	5%	15%	25%	35%	45%
3	0.127	0.395	0.653	0.926	1.389
5	0.127	0.383	0.646	0.918	1.763
10	0.123	0.378	0.657	1.145	2.022
15	0.135	0.419	0.793	1.326	2.142

Table 4: U500 data set: RMSE.

Algorithm	5%	15%	25%	35%	45%
MICE	0.946	1.442	2.247	2.234	3.298
Mean	1.191	2.092	2.754	3.253	3.711
Regression	1.108	1.881	2.499	2.977	3.453
IVIACLR with 2 clusters					
0-0-03-01	0.629	1.125	1.548	1.955	2.864
1-0-03-01	0.641	1.133	1.599	2.096	3.550
0-1-03-01	0.608	1.129	1.592	1.980	2.733
1-1-03-01	0.672	1.123	1.590	2.087	3.497
0-0-05-01	0.620	1.115	1.521	1.984	3.141
1-0-05-01	0.605	1.114	1.571	2.178	3.707
0-1-05-01	0.601	1.093	1.531	1.982	2.999
1-1-05-01	0.592	1.113	1.553	2.118	3.657
0-0-03-05	0.598	1.061	1.364	1.631	2.664
1-0-03-05	0.617	1.105	1.376	1.676	2.784
0-1-03-05	0.619	1.082	1.408	1.663	2.575
1-1-03-05	0.680	1.085	1.397	1.673	2.601
0-0-05-05	0.595	1.031	1.344	1.712	3.258
1-0-05-05	0.572	1.063	1.349	1.618	3.193
0-1-05-05	0.566	1.050	1.343	1.697	3.113
1-1-05-05	0.583	1.061	1.342	1.615	3.223
0-0-03-10	0.608	1.095	1.401	1.626	2.621
1-0-03-10	0.630	1.079	1.410	1.652	2.796
0-1-03-10	0.616	1.090	1.387	1.656	2.538
1-1-03-10	0.665	1.094	1.407	1.666	2.693
0-0-05-10	0.628	1.045	1.337	1.618	3.216
1-0-05-10	0.620	1.056	1.338	1.670	3.140
0-1-05-10	0.597	1.044	1.366	1.662	3.191
1-1-05-10	0.620	1.054	1.351	1.643	3.097
IVIACLR with 3 clusters					
0-0-03-01	0.621	1.094	1.435	1.766	2.618
1-0-03-01	0.635	1.170	1.540	2.000	3.091
0-1-03-01	0.618	1.102	1.475	1.768	2.434
1-1-03-01	0.674	1.166	1.582	2.015	3.039
0-0-05-01	0.607	1.066	1.385	1.745	3.047
1-0-05-01	0.609	1.119	1.518	2.043	3.359
0-1-05-01	0.587	1.049	1.389	1.713	2.716
1-1-05-01	0.595	1.108	1.500	1.987	3.260
0-0-03-05	0.610	1.065	1.375	1.687	2.341
1-0-03-05	0.621	1.089	1.413	1.727	2.470
0-1-03-05	0.639	1.092	1.419	1.685	2.122
1-1-03-05	0.681	1.092	1.438	1.700	2.245
0-0-05-05	0.587	1.027	1.362	1.648	2.691
1-0-05-05	0.583	1.064	1.345	1.640	2.835
0-1-05-05	0.565	1.034	1.354	1.632	2.535
1-1-05-05	0.583	1.049	1.349	1.631	2.644
0-0-03-10	0.605	1.098	1.449	1.687	2.196
1-0-03-10	0.639	1.072	1.419	1.730	2.299
0-1-03-10	0.627	1.090	1.412	1.683	2.207
1-1-03-10	0.674	1.089	1.434	1.724	2.350
0-0-05-10	0.614	1.038	1.363	1.628	2.723
1-0-05-10	0.628	1.055	1.357	1.660	2.663
0-1-05-10	0.598	1.037	1.372	1.628	2.642
1-1-05-10	0.606	1.051	1.360	1.629	2.637

Table 5: U500 data set: MAE.

Algorithm	5%	15%	25%	35%	45%
MICE	0.154	0.416	0.806	0.964	1.651
Mean	0.320	0.969	1.650	2.305	2.976
Regression	0.287	0.843	1.434	1.988	2.546
IVIACLR with 2 clusters					
0-0-03-01	0.133	0.425	0.768	1.166	1.903
1-0-03-01	0.137	0.423	0.780	1.228	2.395
0-1-03-01	0.127	0.424	0.781	1.170	1.830
1-1-03-01	0.137	0.414	0.775	1.220	2.338
0-0-05-01	0.131	0.427	0.767	1.205	2.049
1-0-05-01	0.127	0.425	0.786	1.304	2.503
0-1-05-01	0.127	0.418	0.774	1.196	1.984
1-1-05-01	0.127	0.420	0.775	1.269	2.462
0-0-03-05	0.124	0.383	0.637	0.903	1.710
1-0-03-05	0.126	0.396	0.639	0.916	1.796
0-1-03-05	0.127	0.389	0.653	0.915	1.642
1-1-03-05	0.137	0.391	0.649	0.912	1.688
0-0-05-05	0.125	0.378	0.637	0.995	2.179
1-0-05-05	0.122	0.387	0.640	0.923	2.118
0-1-05-05	0.120	0.385	0.639	0.993	2.078
1-1-05-05	0.124	0.388	0.639	0.914	2.167
0-0-03-10	0.125	0.398	0.651	0.891	1.644
1-0-03-10	0.128	0.389	0.653	0.909	1.785
0-1-03-10	0.125	0.394	0.645	0.910	1.630
1-1-03-10	0.135	0.393	0.650	0.912	1.732
0-0-05-10	0.129	0.384	0.635	0.933	2.137
1-0-05-10	0.128	0.383	0.636	0.952	2.078
0-1-05-10	0.125	0.383	0.648	0.951	2.169
1-1-05-10	0.129	0.381	0.639	0.945	2.056
IVIACLR with 3 clusters					
0-0-03-01	0.133	0.397	0.665	0.965	1.653
1-0-03-01	0.134	0.424	0.723	1.106	1.978
0-1-03-01	0.128	0.398	0.671	0.961	1.536
1-1-03-01	0.139	0.427	0.741	1.109	1.933
0-0-05-01	0.130	0.393	0.654	0.994	1.923
1-0-05-01	0.129	0.420	0.727	1.163	2.195
0-1-05-01	0.125	0.385	0.659	0.967	1.744
1-1-05-01	0.129	0.411	0.714	1.131	2.138
0-0-03-05	0.126	0.384	0.641	0.930	1.496
1-0-03-05	0.128	0.392	0.648	0.940	1.601
0-1-03-05	0.131	0.396	0.662	0.922	1.337
1-1-03-05	0.138	0.391	0.667	0.938	1.458
0-0-05-05	0.125	0.375	0.645	0.930	1.782
1-0-05-05	0.123	0.392	0.633	0.924	1.902
0-1-05-05	0.121	0.380	0.641	0.919	1.677
1-1-05-05	0.124	0.384	0.641	0.925	1.773
0-0-03-10	0.125	0.399	0.668	0.930	1.407
1-0-03-10	0.132	0.389	0.647	0.948	1.448
0-1-03-10	0.127	0.395	0.653	0.926	1.389
1-1-03-10	0.137	0.390	0.657	0.945	1.504
0-0-05-10	0.129	0.383	0.640	0.915	1.806
1-0-05-10	0.131	0.382	0.641	0.930	1.784
0-1-05-10	0.127	0.383	0.646	0.918	1.763
1-1-05-10	0.128	0.382	0.643	0.913	1.738

Table 6: U500 data set: UCE and CCM.

UCE

Algorithm	5%	15%	25%	35%	45%
MICE	5.66	5.96	20.18	20.00	27.56
Mean	27.18	27.00	27.40	26.00	22.10
Regression	20.86	21.18	21.64	21.22	20.34
IVIACLR with 3 clusters	0.08	0.16	1.26	2.08	20.92
IVIACLR with 2 clusters	0.10	1.38	3.44	6.22	30.34

CCM

Algorithm	5%	15%	25%	35%	45%
MICE	0.58	0.64	1.94	3.23	2.60
Mean	2.90	3.14	3.51	3.91	4.75
Regression	2.34	2.83	3.33	3.62	3.87
IVIACLR with 3 clusters	0.05	0.10	0.28	0.43	2.88
IVIACLR with 2 clusters	0.05	0.25	0.58	1.05	4.18

Table 7: D500 data set: RMSE.

Algorithm	5%	15%	25%	35%	45%
MICE	0.699	1.230	1.610	1.870	2.130
Mean	0.590	0.984	1.282	1.521	1.710
Regression	0.588	0.993	1.285	1.527	1.734
IVIACLR					
0-0-03-01	0.609	1.038	1.344	1.592	1.797
1-0-03-01	0.614	1.037	1.364	1.583	1.792
2-0-03-01	0.601	1.046	1.340	1.600	1.792
0-1-03-01	0.619	1.062	1.353	1.596	1.820
1-1-03-01	0.625	1.042	1.358	1.605	1.814
2-1-03-01	0.615	1.065	1.345	1.602	1.806
0-0-05-01	0.579	0.992	1.297	1.519	1.708
1-0-05-01	0.585	0.983	1.289	1.519	1.728
2-0-05-01	0.579	0.992	1.283	1.516	1.739
0-1-05-01	0.587	0.987	1.290	1.520	1.710
1-1-05-01	0.592	0.997	1.279	1.526	1.727
2-1-05-01	0.580	0.986	1.297	1.524	1.726
0-0-03-05	0.608	1.047	1.352	1.631	1.900
1-0-03-05	0.603	1.043	1.387	1.634	1.893
2-0-03-05	0.603	1.048	1.363	1.645	1.900
0-1-03-05	0.613	1.050	1.376	1.652	1.905
1-1-03-05	0.613	1.053	1.375	1.651	1.915
2-1-03-05	0.622	1.050	1.379	1.636	1.915
0-0-05-05	0.573	0.984	1.309	1.586	1.874
1-0-05-05	0.581	1.002	1.303	1.583	1.862
2-0-05-05	0.578	0.990	1.316	1.560	1.859
0-1-05-05	0.577	0.993	1.304	1.583	1.856
1-1-05-05	0.582	1.002	1.320	1.579	1.850
2-1-05-05	0.576	0.994	1.309	1.592	1.860
0-0-03-10	0.616	1.050	1.362	1.629	1.893
1-0-03-10	0.598	1.043	1.351	1.659	1.925
2-0-03-10	0.609	1.039	1.364	1.640	1.921
0-1-03-10	0.612	1.061	1.376	1.646	1.913
1-1-03-10	0.619	1.049	1.375	1.670	1.932
2-1-03-10	0.619	1.035	1.364	1.667	1.923
0-0-05-10	0.585	1.000	1.305	1.592	1.907
1-0-05-10	0.579	0.988	1.307	1.586	1.898
2-0-05-10	0.579	0.993	1.303	1.590	1.901
0-1-05-10	0.581	0.999	1.325	1.594	1.911
1-1-05-10	0.586	0.991	1.320	1.596	1.871
2-1-05-10	0.567	0.991	1.315	1.585	1.907

Table 8: D500 data set: MAE.

Algorithm	5%	15%	25%	35%	45%
MICE	0.244	0.749	1.270	1.750	2.280
Mean	0.223	0.642	1.085	1.515	1.928
Regression	0.224	0.648	1.079	1.525	1.952
IVIACLR					
0-0-03-01	0.225	0.670	1.120	1.570	2.010
1-0-03-01	0.232	0.676	1.140	1.570	2.000
2-0-03-01	0.225	0.676	1.120	1.580	2.000
0-1-03-01	0.230	0.685	1.130	1.570	2.030
1-1-03-01	0.233	0.671	1.130	1.580	2.020
2-1-03-01	0.228	0.687	1.130	1.580	2.020
0-0-05-01	0.219	0.650	1.090	1.520	1.930
1-0-05-01	0.222	0.642	1.080	1.510	1.940
2-0-05-01	0.219	0.647	1.080	1.510	1.960
0-1-05-01	0.223	0.644	1.090	1.520	1.930
1-1-05-01	0.223	0.650	1.080	1.530	1.950
2-1-05-01	0.220	0.643	1.100	1.520	1.940
0-0-03-05	0.230	0.677	1.123	1.604	2.107
1-0-03-05	0.226	0.673	1.152	1.601	2.104
2-0-03-05	0.228	0.674	1.132	1.620	2.103
0-1-03-05	0.231	0.676	1.141	1.628	2.106
1-1-03-05	0.229	0.681	1.143	1.623	2.119
2-1-03-05	0.233	0.676	1.145	1.603	2.121
0-0-05-05	0.219	0.644	1.099	1.566	2.083
1-0-05-05	0.221	0.656	1.093	1.564	2.075
2-0-05-05	0.221	0.643	1.103	1.542	2.065
0-1-05-05	0.218	0.646	1.095	1.565	2.067
1-1-05-05	0.220	0.652	1.106	1.558	2.064
2-1-05-05	0.220	0.650	1.099	1.574	2.074
0-0-03-10	0.232	0.677	1.129	1.598	2.095
1-0-03-10	0.224	0.669	1.122	1.636	2.139
2-0-03-10	0.229	0.671	1.138	1.610	2.123
0-1-03-10	0.231	0.683	1.145	1.619	2.111
1-1-03-10	0.232	0.678	1.375	1.640	2.137
2-1-03-10	0.231	0.667	1.133	1.636	2.130
0-0-05-10	0.224	0.655	1.095	1.571	2.117
1-0-05-10	0.222	0.645	1.097	1.565	2.112
2-0-05-10	0.221	0.648	1.094	1.572	2.114
0-1-05-10	0.221	0.655	1.111	1.574	2.127
1-1-05-10	0.223	0.647	1.110	1.580	2.078
2-1-05-10	0.216	0.647	1.105	1.560	2.117

Table 9: D500 data set: UCE and CCM.

UCE

Algorithm	5%	15%	25%	35%	45%
MICE	37.54	44.52	52.06	59.60	62.66
Mean	27.26	39.18	52.90	55.48	60.78
Regression	33.00	33.68	52.82	54.70	62.88
IVIACLR	29.48	37.94	52.38	52.50	59.14

CCM

Algorithm	5%	15%	25%	35%	45%
MICE	2.85	3.04	3.20	2.79	3.47
Mean	2.42	2.84	2.60	3.13	2.90
Regression	3.22	2.71	3.24	2.61	2.95
IVIACLR	2.60	3.55	2.66	3.54	3.06

Table 10: U2500 data set: RMSE, MAE, UCE, and CCM.

RMSE

Algorithm	5%	15%	25%	35%	45%
MICE	2.637	4.811	6.933	12.396	26.818
Mean	27.499	47.618	61.491	72.789	82.535
Regression	27.575	60.883	121.937	1264.698	1370.102
IVIACLR	2.161	3.767	4.974	7.176	18.565

MAE

Algorithm	5%	15%	25%	35%	45%
MICE	1.897	5.723	9.715	14.954	26.802
Mean	23.502	70.486	117.534	164.503	211.484
Regression	21.666	84.701	216.796	3159.035	3919.392
IVIACLR	1.628	4.913	8.269	12.188	20.426

UCE

Algorithm	5%	15%	25%	35%	45%
MICE	0.00	0.00	0.00	0.03	0.22
Mean	0.00	0.00	0.02	2.47	16.95
Regression	0.00	2.50	23.19	76.46	74.85
IVIACLR	0.00	0.00	0.00	0.01	0.29

CCM

Algorithm	5%	15%	25%	35%	45%
MICE	0.16	0.31	0.52	1.28	4.24
Mean	25.47	76.40	127.46	663.71	590.11
Regression	20.38	332.43	623.45	3290.55	4063.28
IVIACLR	0.14	0.27	0.41	0.82	2.10

Table 11: Iris data set: RMSE.

Algorithm	5%	15%	25%	35%	45%
MICE	0.167	0.305	0.448	0.522	0.651
Mean	0.471	0.828	1.095	1.282	1.440
Regression	0.960	1.965	2.780	3.870	4.807
IVIACLR					
0-0-03-01	0.145	0.281	0.452	0.674	0.963
1-0-03-01	0.215	0.570	0.906	1.920	1.730
2-0-03-01	0.215	0.557	0.874	1.500	1.560
0-1-03-01	0.137	0.279	0.452	0.686	0.956
1-1-03-01	0.214	0.595	0.949	1.370	1.810
2-1-03-01	0.203	0.559	0.902	1.530	1.590
0-0-05-01	0.136	0.282	0.450	0.663	0.881
1-0-05-01	0.220	0.580	0.940	1.749	1.701
2-0-05-01	0.218	0.594	0.961	1.444	1.752
0-1-05-01	0.142	0.282	0.440	0.646	0.886
1-1-05-01	0.221	0.572	0.980	1.330	1.769
2-1-05-01	0.213	0.565	0.953	1.555	1.711
0-0-03-05	0.131	0.266	0.408	0.651	0.919
1-0-03-05	0.144	0.366	0.581	1.270	1.260
2-0-03-05	0.144	0.370	0.616	1.200	1.180
0-1-03-05	0.131	0.262	0.403	0.655	0.901
1-1-03-05	0.144	0.390	0.623	1.070	1.230
2-1-03-05	0.141	0.350	0.632	1.260	1.240
0-0-05-05	0.129	0.270	0.412	0.641	0.853
1-0-05-05	0.142	0.352	0.620	1.036	1.233
2-0-05-05	0.145	0.356	0.680	1.211	1.229
0-1-05-05	0.128	0.268	0.401	0.620	0.856
1-1-05-05	0.145	0.344	0.594	0.922	1.332
2-1-05-05	0.140	0.356	0.616	1.285	1.257
0-0-03-10	0.135	0.272	0.407	0.651	0.929
1-0-03-10	0.143	0.328	0.497	1.010	1.200
2-0-03-10	0.142	0.317	0.552	1.080	1.140
0-1-03-10	0.131	0.260	0.403	0.659	0.923
1-1-03-10	0.138	0.334	0.523	0.928	1.210
2-1-03-10	0.140	0.318	0.528	1.200	1.200
0-0-05-10	0.130	0.273	0.415	0.650	0.866
1-0-05-10	0.138	0.315	0.509	0.919	1.169
2-0-05-10	0.138	0.315	0.556	1.100	1.153
0-1-05-10	0.130	0.273	0.400	0.618	0.862
1-1-05-10	0.136	0.317	0.501	0.819	1.239
2-1-05-10	0.138	0.315	0.503	1.183	1.193

Table 12: Iris data set: MAE.

Algorithm	5%	15%	25%	35%	45%
MICE	0.054	0.176	0.332	0.460	0.641
Mean	0.163	0.495	0.836	1.160	1.478
Regression	0.308	1.124	2.161	3.661	5.082
IVIACLR					
0-0-03-01	0.049	0.159	0.314	0.533	0.849
1-0-03-01	0.060	0.255	0.563	1.170	1.500
2-0-03-01	0.061	0.254	0.547	1.080	1.390
0-1-03-01	0.047	0.158	0.311	0.539	0.837
1-1-03-01	0.060	0.267	0.592	1.040	1.580
2-1-03-01	0.057	0.251	0.572	1.100	1.440
0-0-05-01	0.047	0.156	0.312	0.525	0.803
1-0-05-01	0.062	0.259	0.584	1.172	1.500
2-0-05-01	0.061	0.264	0.600	1.064	1.531
0-1-05-01	0.048	0.156	0.305	0.519	0.807
1-1-05-01	0.061	0.255	0.607	1.010	1.547
2-1-05-01	0.060	0.254	0.598	1.136	1.502
0-0-03-05	0.045	0.151	0.284	0.500	0.790
1-0-03-05	0.047	0.178	0.358	0.774	1.030
2-0-03-05	0.047	0.180	0.374	0.858	0.994
0-1-03-05	0.045	0.148	0.279	0.504	0.784
1-1-03-05	0.047	0.184	0.380	0.762	1.050
2-1-03-05	0.046	0.172	0.387	0.898	1.030
0-0-05-05	0.044	0.150	0.284	0.500	0.761
1-0-05-05	0.047	0.171	0.377	0.713	1.020
2-0-05-05	0.048	0.174	0.397	0.858	1.059
0-1-05-05	0.044	0.148	0.279	0.488	0.763
1-1-05-05	0.047	0.169	0.357	0.679	1.110
2-1-05-05	0.046	0.173	0.382	0.891	1.085
0-0-03-10	0.046	0.153	0.285	0.502	0.800
1-0-03-10	0.047	0.168	0.316	0.677	0.987
2-0-03-10	0.047	0.163	0.339	0.767	0.946
0-1-03-10	0.044	0.147	0.279	0.508	0.792
1-1-03-10	0.046	0.168	0.328	0.676	1.020
2-1-03-10	0.046	0.162	0.330	0.849	0.985
0-0-05-10	0.045	0.150	0.283	0.501	0.770
1-0-05-10	0.046	0.161	0.317	0.642	0.955
2-0-05-10	0.045	0.162	0.339	0.773	0.976
0-1-05-10	0.044	0.150	0.277	0.489	0.766
1-1-05-10	0.045	0.163	0.311	0.599	1.020
2-1-05-10	0.046	0.162	0.319	0.812	0.999

Table 13: Iris data set: UCE and CCM.

UCE

Algorithm	5%	15%	25%	35%	45%
MICE	1.07	2.67	6.20	6.73	8.60
Mean	5.80	16.53	25.60	28.87	32.87
Regression	24.13	34.33	43.67	53.33	52.20
IVIACLR	0.67	3.27	7.53	10.60	13.40

CCM

Algorithm	5%	15%	25%	35%	45%
MICE	0.04	0.12	0.24	0.26	0.34
Mean	0.22	0.63	0.82	0.98	0.70
Regression	1.84	4.50	6.37	5.71	4.90
IVIACLR	0.04	0.12	0.23	0.18	0.28

Table 14: TSPLIB data set: RMSE, MAE, UCE, and CCM.

RMSE

Algorithm	5%	15%	25%	35%	45%
MICE	1618.654	2783.380	3736.581	4575.201	5126.254
Mean	1151.442	2002.080	2638.503	3176.327	3648.815
Regression	1149.538	2003.418	2644.599	3205.853	3728.615
IVIACLR	1204.708	2115.541	2835.145	3466.203	4084.690

MAE

Algorithm	5%	15%	25%	35%	45%
MICE	380.063	1130.523	1982.582	2865.145	3673.430
Mean	288.754	868.460	1488.416	2133.230	2789.135
Regression	291.856	879.474	1503.643	2161.930	2856.578
IVIACLR	294.381	895.796	1549.340	2264.830	3037.619

UCE

Algorithm	5%	15%	25%	35%	45%
MICE	6.44	20.70	29.69	47.35	55.03
Mean	10.94	28.20	46.25	52.25	55.41
Regression	14.90	27.62	46.99	51.29	56.25
IVIACLR	8.11	22.08	40.69	53.01	57.01

CCM

Algorithm	5%	15%	25%	35%	45%
MICE	142.15	841.74	598.68	2150.13	2891.26
Mean	416.60	2225.65	4878.25	5213.65	5930.32
Regression	601.82	1102.69	3831.45	4721.49	5769.44
IVIACLR	250.37	709.27	1953.15	3104.19	3381.53

Table 15: Red wine quality data set: RMSE, MAE, UCE, and CCM.

RMSE					
Algorithm	5%	15%	25%	35%	45%
MICE	7.450	13.600	18.300	23.000	26.800
Mean	5.635	10.739	14.363	18.202	21.023
Regression	3003.404	1450.330	668.012	660.305	50.921
IVIACLR					
0-1-05-05	5.523	10.770	14.527	18.278	22.378
0-1-05-10	5.495	10.877	14.584	19.992	23.312

MAE					
Algorithm	5%	15%	25%	35%	45%
MICE	1.670	5.350	9.120	13.700	18.000
Mean	1.295	4.156	7.103	10.790	14.375
Regression	103.783	60.521	45.281	74.159	22.404
IVIACLR					
0-1-05-05	1.262	4.133	7.220	11.084	15.361
0-1-05-10	1.236	4.176	7.308	12.038	15.912

UCE					
Algorithm	5%	15%	25%	35%	45%
MICE	4.14	12.86	20.28	31.63	39.01
Mean	3.41	16.60	22.96	33.01	39.51
Regression	51.61	58.56	56.56	58.88	44.05
IVIACLR					
0-1-05-05	3.64	10.73	25.62	39.32	41.78
0-1-05-10	3.83	12.51	26.74	41.28	42.58

CCM					
Algorithm	5%	15%	25%	35%	45%
MICE	1.27	10.99	20.57	34.70	24.76
Mean	0.92	15.82	39.49	30.88	61.40
Regression	26666.92	12092.51	7632.57	6540.85	337.75
IVIACLR					
0-1-05-05	0.90	3.79	51.76	59.14	77.12
0-1-05-10	1.13	12.44	36.25	69.44	68.09

Table 16: Abalone: RMSE, MAE, UCE, and CCM.

RMSE

Algorithm	5%	15%	25%	35%	45%
MICE	0.713	1.251	1.620	1.955	2.307
Mean	0.751	1.268	1.632	1.937	2.200
Regression	3.073	5.286	7.179	10.825	14.102
IVIACLR					
0-1-05-05	0.532	0.912	1.230	1.660	2.024
0-1-05-10	0.532	0.901	1.245	1.675	1.941

MAE

Algorithm	5%	15%	25%	35%	45%
MICE	0.126	0.380	0.642	0.921	1.246
Mean	0.173	0.507	0.838	1.175	1.510
Regression	1.143	3.194	5.097	7.984	11.268
IVIACLR					
0-1-05-05	0.095	0.278	0.475	0.734	1.026
0-1-05-10	0.094	0.275	0.478	0.758	1.013

UCE

Algorithm	5%	15%	25%	35%	45%
MICE	2.36	7.06	7.48	10.24	13.01
Mean	6.38	7.28	4.69	5.87	7.85
Regression	12.46	20.71	21.01	20.03	24.32
IVIACLR					
0-1-05-05	6.96	10.37	10.68	13.33	16.60
0-1-05-10	8.70	10.25	10.76	13.43	15.91

CCM

Algorithm	5%	15%	25%	35%	45%
MICE	0.16	0.41	0.27	0.40	0.42
Mean	0.53	0.51	0.32	0.39	0.54
Regression	5.47	8.43	10.25	13.68	18.30
IVIACLR					
0-1-05-05	0.65	0.91	0.91	1.03	1.20
0-1-05-10	0.85	0.98	1.00	1.08	1.27

Table 17: White wine quality data set: RMSE, MAE, UCE, and CCM.

RMSE

Algorithm	5%	15%	25%	35%	45%
MICE	10.216	18.179	24.618	29.928	35.256
Mean	7.664	14.117	19.356	23.920	27.928
Regression	438.435	14641.733	873.038	2153.377	2443.363
IVIACLR	7.640	14.455	19.058	23.936	28.104

MAE

Algorithm	5%	15%	25%	35%	45%
MICE	2.628	8.181	14.120	20.252	27.020
Mean	1.907	6.207	10.911	16.015	21.223
Regression	19.248	326.706	32.080	91.620	304.430
IVIACLR	1.846	5.991	10.470	15.737	21.173

UCE

Algorithm	5%	15%	25%	35%	45%
MICE	20.02	31.76	39.61	55.65	51.90
Mean	22.03	30.35	43.52	50.82	56.41
Regression	52.71	65.44*	59.94	65.79	73.18*
IVIACLR	24.74	30.89	44.04	49.19	54.65

CCM

Algorithm	5%	15%	25%	35%	45%
MICE	26.88	40.08	48.69	78.20	39.44
Mean	45.71	66.03	81.50	99.82	98.18
Regression	6577.84	77841.85*	11243.11	31164.36	13617.70*
IVIACLR	49.97	54.88	73.38	116.02	91.50

* averaged over 9 runs.

Turku
Centre *for*
Computer
Science

Joukahaisenkatu 3-5 A, 20520 TURKU, Finland | www.tucs.fi

University of Turku
Faculty of Mathematics and Natural Sciences
- Department of Information Technology
- Department of Mathematics and Statistics

Turku School of Economics
- Institute of Information Systems Sciences

Åbo Akademi University
- Computer Science
- Computer Engineering

ISBN 978-952-12-3689-1
ISSN 1239-1891