Missing Value Imputation via Clusterwise Linear Regression

Napsu Karmitsa, Sona Taheri, Adil Bagirov, and Pauliina Mäkinen

***Abstract*—In this paper a new method of preprocessing incomplete data is introduced. The method is based on clusterwise linear regression and it combines two well-known approaches for missing value imputation: linear regression and clustering. The idea is to approximate missing values using only those data points that are somewhat similar to the incomplete data point. A similar idea is used also in clustering based imputation methods. Nevertheless, here the linear regression approach is used within each cluster to accurately predict the missing values, and this is done simultaneously to clustering. The proposed method is tested using some synthetic and real-world data sets and compared with other algorithms for missing value imputations. Numerical results demonstrate that this method produces the most accurate imputations in MCAR and MAR data sets with a clear structure and the percentages of missing data no more than 25%.**

***Index Terms*—Data analysis, Incomplete data, Imputation, Clusterwise linear regression, Nonsmooth optimization.**

I. Introduction

THE occurrence of missing (or incomplete) data is very common in many fields of research such as social sciences, biology, medicine and climate science. There are various reasons for possible incompleteness of data. For instance, in medical domain some data may be missing due to not performing certain procedures on a given patient or due to malfunction of a certain equipment. In addition, data may be missing because the patient chose not to disclose them [1].

As the quality of knowledge extracted from a data depends strongly on the quality of the data, missing values may have a significant effect on the conclusions drawn from the data. Moreover, most of the existing knowledge discovery and data mining algorithms, used for example in clustering and classification, are designed under the assumption that there are no missing values in the data. The performance of these algorithms may be considerably worsened when data is incomplete. Therefore, data preprocessing is a critical task in the knowledge discovery to ensure the quality of the data to be analyzed and the performance of the tools to be used.

In this paper, a new missing value imputation method, IVIACLR (*Imputation via Clusterwise Linear Regression*), is introduced where clusterwise linear regression (CLR) is used to predict suitable imputations. In CLR clustering is intertwined with regression. Thus, the "clusters" formed with CLR are not clusters in the traditional ball-shaped form, and the imputation results (predictions) obtained with CLR differ significantly from those obtained with its closest counterpart [2], [3] where a combination of clustering — with some cluster centers and ball-shaped clusters — and linear regression is used for missing value imputation. This makes CLR unique technique to predict the missing values. To the best of our knowledge, this is the first time the CLR approach is used for missing value imputation.

The main idea in the IVIACLR is to use those data points for regression which are similar to the incomplete data object. More precisely, we determine the value of a missing feature based on the object's observed features and its similarity to other objects in the data set. The difference between the IVIACLR and the methods [2], [3] mentioned above is that, instead of using clustering and regression separately one after another, we apply the CLR (with three different prediction methods) to predict suitable imputations. Therefore, "cluster centers" in the IVIACLR are linear regression functions and the IVIACLR computes all the predictions to missing values simultaneously to clustering. In addition, the IVIACLR applies the incremental approach, which allows us to utilize intermediate results and, thus, knowing the number of clusters a priori is not as crucial as in clustering based approaches.

The performance of the proposed method is studied and compared to other imputation methods using three artificial and five real-world data sets of various sizes with varying percentages of missing values. Most of our experiments are made under the MCAR assumption. In addition, some tests are run with the MAR and MNAR missing data mechanisms (see e.g. [4], [5]). The evaluation metrics 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 displacement* (CCD) criterion that can be used together with the UCE to measure bias in the imputed values.

The main contributions of the paper are to

- introduce a novel CLR based approach for missing value imputation;
- implement a new algorithm IVIACLR to impute missing values (available at GitHub https://github.com/SnTa2019/Missing-Value-Imputation);
- propose a new CCD criterion.

The rest of the paper is organized as follows. Section II provides a brief overview of existing methods for dealing with incomplete data. The CLR problem and an algorithm for its solution are described in Section III. Section IV introduces the IVIACLR. The new CCD criterion is introduced and the results of numerical experiments are reported in Section V. Finally, Section VI concludes the paper.

N. Karmitsa and P. Mäkinen are with Department of Mathematics and Statistics, University of Turku, FI-20014 Turku, Finland; e-mail: napsu@karmitsa.fi

S. Taheri and A. Bagirov are with School of Science, Engineering and Information Technology, Federation University Australia, Victoria, Australia.

II. Related work

To date, there are several approaches to deal with incomplete data. These approaches can be divided into three main categories:

1) *Deletion-based methods* (e.g. pairwise deletion and listwise deletion [4], [5]) strive toward the complete data by removing all the observations/attributes containing missing values. Because of their simplicity, these methods are fairly popular. However, they may lead to a significant loss of information which call for thorough consideration before using them [4].
2) *Learning methods for complete and incomplete data* apply machine learning techniques to classify or cluster incomplete data directly without explicitly estimating missing features or modifying the data set. These methods include those based on clustering methods like modifications of k-means [6]–[9] and fuzzy c-means [10], [11], neural networks [12]–[16] and different variants of kernel methods [17]–[19].
3) *Imputation methods* fill missing values in order to complete the original data set without significant loss of information. The key advantage of these methods is their ability to create a complete data set by embedding new values (predictions) without changing the original observed values in the data set.

Imputation methods, in their turn, can be divided into three groups:

1) *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 [4], [20].
2) *Model-based imputation methods* use mathematical or statistical models to handle missing values and to predict correct imputations. This group consists mainly of regression and maximum likelihood based algorithms like the multiple imputations by chained equations (MICE) and stochastic regression [4], [21]–[23], and expectation-maximization (EM) [4], [24], [25].
3) *Machine learning based algorithms* include those which use neural networks [26], [27], clustering and classification [28]–[31], and k-nearest neighbours (k-NN) [32], [33] techniques.

III. Background

A. 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 $\boldsymbol{a}^T\boldsymbol{b}$ (bolded symbols are used for vectors).

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

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

B. Nonsmooth Optimization

Nonsmooth optimization (NSO) refers to the general problem of minimizing (or maximizing) functions which are not continuously differentiable. This means that the gradient $\nabla f(\boldsymbol{x})$ of the function f needs not to exist for all $\boldsymbol{x} \in \mathbb{R}^n$. However, we can define the subdifferential [34] 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 at a point $\boldsymbol{x} \in \mathbb{R}^n$ is given by

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

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 NSO, we refer to [35].

C. Clusterwise Linear Regression

CLR is a technique to approximate data using two or more linear functions. It is based on two well-known approaches: clustering and regression. Given a data set $A = \{(\boldsymbol{a}_i, b_i) \in \mathbb{R}^{n-1} \times \mathbb{R} \mid \ i = 1, \dots, m\}$, CLR aims to partition it into k clusters and, simultaneously, to find regression coefficients $\{\boldsymbol{x}_j, y_j\}$, $\boldsymbol{x}_j \in \mathbb{R}^{n-1}$, $y_j \in \mathbb{R}$, $j = 1, \dots, k$ within clusters in order to minimize the overall fit. Let $\emptyset \neq A^j \subset A$, $j = 1, \dots, k$ be clusters such that

1. $A^j \bigcap_k A^l = \emptyset$, for all $j, l = 1, \dots, k$, $j \neq l$, and
2. $A = \bigcup_{j=1} A^j$.

Let $\{\boldsymbol{x}_j, y_j\}$ be linear regression coefficients computed using solely the data points from the cluster A^j, $j = 1, \dots, k$. Then for a given 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. The function

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

is called the *k-th CLR function* (*overall fit function*) [36]–[38]. Here $\boldsymbol{x} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_k) \in \mathbb{R}^{(n-1)k}$ and $\boldsymbol{y} = (y_1, \dots, y_k) \in \mathbb{R}^k$. The *NSO formulation of the CLR problem* is given by

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

Problem (1) is convex for $k = 1$ and nonsmooth, nonconvex, and piecewise quadratic for $k > 1$. The number of clusters k is not always known a priori and it 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 objects in a data set.

CLR has many applications (see, e.g. [39]–[44]). The algorithms for solving general CLR problems are those based on data mining [45]–[47]; statistical [43], [48], [49]; and optimization approaches [36]–[38], [50]–[53]. In principle, any of these algorithms could be used in the IVIACLR. In this paper, we use the algorithm LMBM-CLR [53] to solve the CLR problem. The reason for choosing this method is that it is as accurate as most other methods but it requires less CPU time (see [53] for numerical comparison of the LMBM-CLR and other CLR algorithms). This is very important as both the incomplete data set and the number of missing values in the data may be large. In addition, the LMBM-CLR is an incremental algorithm, which allows us to use also the intermediate results of the CLR procedure. That is, in addition to the k-th CLR problem (1), the LMBM-CLR solves also all the intermediate l-th CLR problems, where $l = 1, \ldots, k-1$.

D. LMBM-CLR -*Method*

The LMBM-CLR consists of two parts: an *incremental algorithm* is applied to solve CLR problems globally and at each iteration of this algorithm the NSO solver, *limited memory bundle method* (LMBM), is used to solve (locally) both the CLR problem (1) and the so-called *auxiliary CLR problem* starting from various initial points provided by the incremental algorithm. Figure 1 illustrates the structure of this combination. The auxiliary CLR problem has a simple structure and its solutions provide the most promising initial points for the CLR problem when combined with the solution obtained at the previous iteration of the incremental algorithm. At the final stage the data points are associated to the closest regression functions (missing values are not used when computing the regression error).

The underlying optimization solver LMBM is originally developed for solving general large-scale NSO problems. It is a *bundle method* that collects subgradient information from the previous iterations into a bundle to approximate the subdifferential of the objective function. The bundle is used to generate a better model of the objective function if the descent condition, i.e. the serious step, is not satisfied (see Figure 1).

For more details on the LMBM-CLR we refer to [53], on the incremental algorithm and the auxiliary problem to [36], and on the original LMBM to [54], [55].

IV. A CLR ALGORITHM FOR MISSING VALUE IMPUTATION

In this section, we introduce a new imputation method IVIACLR. The IVIACLR consists of three different parts: initial imputation, CLR-method, and prediction. We first introduce the main algorithm and then provide some discussions on the initial imputations and the prediction methods.

A. Main Algorithm

The IVIACLR consists of the inner and outer iterations. In the inner iteration, we go through all features with missing values iteratively using some initial imputations (or previously imputed values) in place of missing values on input variables. We repeat this process (outer iteration) with imputed values as place holders until the results are not changing significantly or the maximum number of outer iterations is reached. The IVIACLR -algorithm is as follows:

Algorithm 1: IVIACLR

Data: An incomplete data set $A = \{\hat{\boldsymbol{a}}_1, \hat{\boldsymbol{a}}_2, \ldots, \hat{\boldsymbol{a}}_m\}$, the maximum number of linear functions k, the maximum number of outer iterations $o_{max} \geq 1$, and a tolerance $\varepsilon \geq 0$.
Result: The 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 predict missing values for the feature j;
    Replace missing values at the feature j with their *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 the current imputed data set.
  Set all originally missing values again as "missing";
STOP with the current imputed data set;

REMARK 1. *The maximum number of linear functions (clusters) is data specific and it should be given as an input parameter to the* IVIACLR. *When an incremental CLR algorithm like* LMBM-CLR *is used within the* IVIACLR *it is enough to give an upper limit and any intermediate number of clusters can be chosen as a solution. The selection can be done, for instance, based on the CLR function values and their decrease when one cluster is added. Possible procedures for "intelligent" stopping will be studied in the future.*

REMARK 2. *In the current version of the* IVIACLR *we only consider continuous numeric data. Discrete (integer) data can be considered simply by rounding the final results. Furthermore, it is possible to generalize the proposed approach to deal with different data types (e.g. binary).*

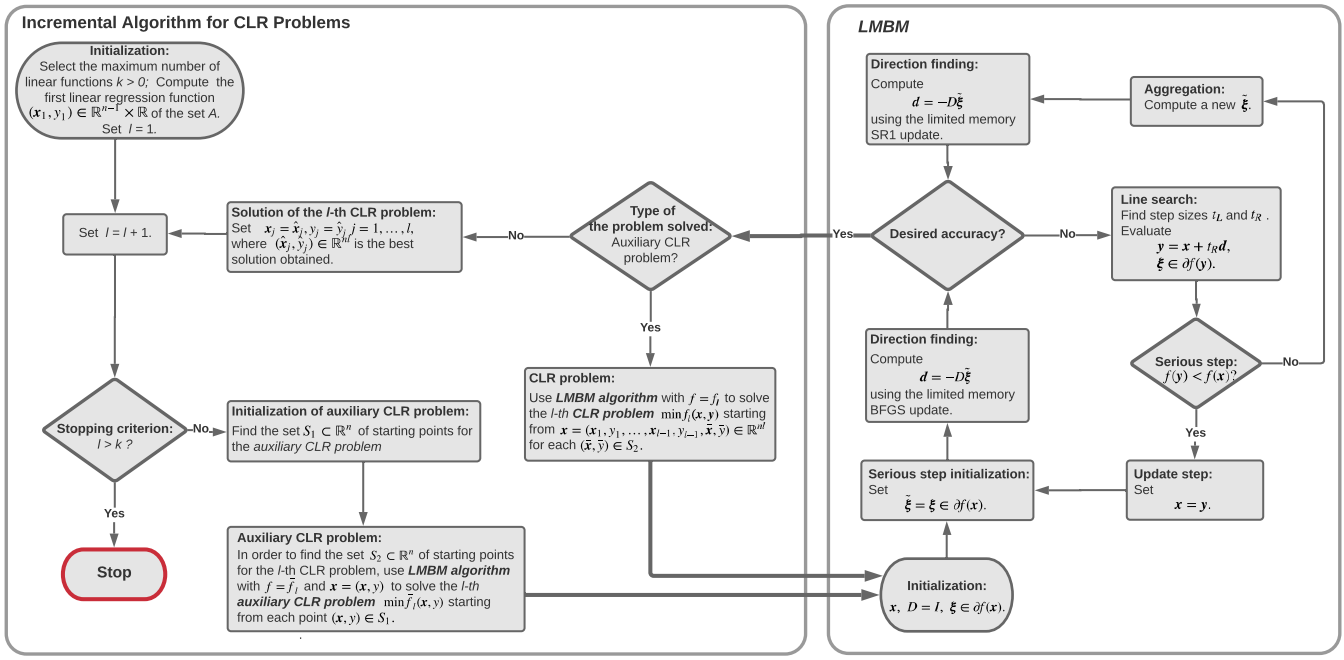


Fig. 1. LMBM-CLR -method. In the LMBM, D denotes the inverse variable metric approximation of the Hessian and $\boldsymbol{\xi} \in \partial f(\boldsymbol{x})$ is an arbitrary element (*subgradient*) from the subdifferential. The flowchart is drawn using Lucidchart (http://www.lucidchart.com).

B. 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 complete data set, i.e. data set produced with deletion, is used and this data is the same for all features with missing values.
3) *Recursive regression imputation.* In the recursive regression imputation the feature with the least number of missing values is imputed first using the complete data. Then the feature with the next fewest missing values is imputed using the updated data set with previously imputed values. This process is repeated until all missing values are imputed.

As a sole imputation method, the mean is known to produce imputations with a high level of bias by pulling or pushing the distribution of the imputed data toward the mean of the observed data. The positive point is that the mean imputation is extremely easy to compute and we can apply it even if every object in the data set has missing values. Thus, it is well suited to be an initial imputation method. On the other hand, the linear (recursive) regression imputation underestimates the variance of imputed items. This is not a major issue in our method due to the fact that more than one linear function is computed in this method. Note that one 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.

C. Prediction methods

Design of a prediction method based on 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 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 a 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 methods are used in the IVIACLR to compute the weight w_j of the j-cluster, $j = 1, \ldots, k$.

1) *Simple weighting method [39].* 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 points in a data set.
2) *Local weighting method.* In the local weighting method we, instead of all m data points, use only $l < m$ nearest neighbours to compute the weight. Once the nearest neighbours have been selected the weight w_j is computed as $w_j = l_j/l$, where l_j is the number of nearest neighbour points from the j-th cluster.
3) *RMSE based local weighting.* In this method, using the RMSE, we first determine how similar the data point $(\boldsymbol{a}_i, b_i)$ with a missing value in b_i is to l of its nearest neighbours $(\boldsymbol{a}_h, b_h)$, $h \in \{1, \ldots, m\}, h \neq i$. Let $\mathcal{C}_{nn}$ be a set of indices of the nearest neighbours (i.e. $h \in \mathcal{C}_{nn}$ and $|\mathcal{C}_{nn}| = l$) and $\mathcal{C}_j$ be a set of indices of the nearest neighbour points from the j-th cluster ($|\mathcal{C}_j| = l_j$), $j =$

$1, \ldots, k$. The weight w_j is computed as

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

where $\overline{rmse} = \sum_{h \in \mathcal{C}_{nn}} rmse_{hi}$, and $rmse_{hi}$ is the RMSE between $\boldsymbol{a}_i$ and $\boldsymbol{a}_h$. Here, we set $w_j = l_j/l$ if $\overline{rmse} = 0$. If $l = 1$, we simply take the nearest neighbour (the one with the smallest $rmse_{hi}$) and the cluster j^* it belongs to, 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.

Although the first approach is the most simple to implement it suffers the same drawback than just a linear regression imputation 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 obtained cluster structure. The problem with these approaches in large data sets is the computational burden when computing the 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 through when seeking for the nearest neighbours.

V. Numerical Experiments

The proposed algorithm IviaCLR is tested using some artificial and real-world data sets and compared to some commonly used methods for imputation: the mean imputation, regression imputation, and MICE [21], [23], [56].

The IviaCLR is implemented in Fortran 2003 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 [57].

Remark 3. *The source code of the* IviaCLR *is available at GitHub (https://github.com/SnTa2019/Missing-Value-Imputation). To use the code one does not need to know Fortran: just giving the input (name of the incomplete data set and its numbers of points and features), the output (name of the imputed data set), and the maximum number of regression functions is enough, and the program will deal with the rest.*

A. Data Sets

To test and compare the above mentioned imputation methods we use three artificial and five real-world data sets. Most of our tests are made with the MCAR data. That is, we generate incomplete data sets with varying percentages of missing values (from 5% to 45% of all entries) by randomly removing some values of the original complete data. Nevertheless, every data point has to have at least one reference attribute in it. In addition, some tests are run with the MAR and MNAR missing data mechanisms. To generate MAR and MNAR data, we apply the multivariate amputation procedure [58] implemented as the function `ampute` [59] in R-package MICE. We use the default parameters of `ampute` and only generate 5, 15 and 25 percentages of missingness due to the limitation of this function when the percentage of missing values is considered with respect to entries of data (mn pieces). For all original data sets we perform 10 runs with all percentages of missing values (and different missing data mechanisms). That is, in 10 runs the original complete data and the percentage of missing values are the same but different values are missing. The final results are averaged over these 10 runs.

a) Artificial Data: To demonstrate the performance of the proposed method in different types of data, the simple synthetic data sets are generated to be very different from each other. The details of these data sets are given in Table I, where m is the number of observations, n is the number of features and k is the optimal number of clusters. All the artificial data used in our experiments are available at https://github.com/SnTa2019/Missing-Value-Imputation.

TABLE I
Artificial data sets.

Data	m	n	k	Structure
D500	500	4	–	no[1]
U500	500	2	3	very clear (clusters)[2]
U10000	10000	20	5	clear (clusters)[3]

[1] Uniform distribution within $[-2, 2]$. [2] See Figure 2.
[3] Slightly overlapping clusters, feature values between 0 and 100.

b) Real-World Data: We use two small and three medium sized real-world data sets in our experiments. These data sets are available from [60]. Their brief description is given in Table II.

TABLE II
Real-world data sets.

Data	m	n	k	Structure
Iris plant	150	4	3	clear (clusters)
Wine recognition	178	13	3	very clear (clusters)
TSPLIB1060	1060	2	5	not very clear (clusters)
Red wine quality	1599	11	6[1]	clear (hyperplanes)
Abalone	4177	8	2[2]	no

[1] Number of clusters in [53]. [2] Number of clusters in [30].

B. Evaluation metrics

Imputation methods are compared using four evaluation metrics:

1) *Root mean square error* (RMSE) measures the difference between the 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 absolute magnitude of the errors and is computed by

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

3) *Unsupervised classification error* (UCE) assesses the preservation of an internal structure. More precisely, 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 [61] as a clustering method (it is available at http://napsu.karmitsa.fi/clustering/).

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

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

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

Note that the sole CCD does not give much information about the accuracy of the imputation. Nevertheless, it supports and reinforces the UCE by indicating if the centers of clusters are also preserved:

- *a small* UCE *and small* CCD: the imputation can be considered accurate;
- *a small* UCE *and large* CCD: there is some bias in imputed values;
- *a large* UCE *and small* CCD: the overall structure of the data set is preserved but some imputed values are incorrect; and
- *both the* UCE *and* CCD *large*: the structure of the data set is lost.

Both the MAE and RMSE measure the average magnitude of the error. The MAE is a linear score which means that all the individual differences are weighted equally on the average. On the other hand, the RMSE is a quadratic score and the errors are squared before they are averaged. Therefore, the RMSE gives a relatively high weight to large errors and it is useful when they are particularly undesirable.

C. Results

We visualise the most relevant results in Figures 3–20 and draw some conclusions. The more comprehensive experiments with justification to our parameter choices can be found in the research report [62].

1) Choice of parameters for IVIACLR*:* There are several parameters that can be adjusted in the IVIACLR: that is, for instance, the type of an initial imputation, the number of outer iterations, the type of a prediction method, and the number of nearest neighbours in the prediction phase. The selected parameters are given in Table III.

TABLE III
PARAMETERS OF IVIACLR.

Parameter	"default value"
maximum no. clusters	data specific, see Remarks 1 and 4
initial imputation	*mean*
prediction	*RMSE based local weighting*
o_{max}	5 (large data), 10 (small data)
l	5
l_{max}	150
ε	not applicable

REMARK 4. *The number of clusters is data specific. However, as noted in Remark 1 due to the incremental approach it is enough to give an upper limit and any intermediate numbers of clusters can be used to produce the solution. In addition, the use of k "too big" does not significantly impact on the accuracy of the imputation (but it increases computational time). That is, making too many regression functions just adds computational cost but the missing values are still imputed to (or near to) the "correct" hyperplane in the prediction phase.*

In the first implemantation of the IVIACLR we do not have the tolerance ε for the change (see Algorithm 1) and we set $l_{max} = 150$ (see Subsection IV-C) for all data sets.

A large number of outer iterations, o_{max} in Algorithm 1, may prevent a "bad" initial solution. Nevertheless, it also adds computational cost. Therefore, we select $o_{max} = 10$ for small data sets, where the cost is insignificant, and $o_{max} = 5$ for larger data sets, to reduce the possible computational burden.

2) Performance of IVIACLR *on synthetic data:* The synthetic data sets D500, U500 and U10000 have different number of points and features and two of them (U500 and U10000) have clear structures while D500 has no structure. We compute three clusters in U500 and five clusters in D500 and U10000.

The original data set U500 as well as the imputed data sets with 5% and 45% of missing values are illustrated in Figures 2–4, respectively. The RMSE, MAE, UCE, and CCD for different imputation methods are given in Figure 5. Recall that RMSE, MAE, UCE and CCD are averaged over 10 runs with different, but the same amount of, missing values. Naturally, all the algorithms are applied to data sets with the same missing values. From the figures we see the superiority of the IVIACLR when data is clearly structured. With 5% of missing values it significantly misplaces only one feature (see Figure 3(a)). This displacement is due to the fact that l neighbours used for prediction belong to different clusters.

Furthermore, the structure of the data is still noticeable in the imputed data set when almost half of the data is missing (see Figure 4(a)). In addition, MICE performs quite well with 5% of missing data. It misplaces the same feature misplaced by the IVIACLR. However, the error with MICE is greater than that of the IviaCLR. From Figure 5 we see that the IVIACLR always produces imputations that have the smallest RMSEs and their MAEs are similar to those produced by MICE. Furthermore, the UCE and CCD show that the IVIACLR is clearly the best in preserving the original structure of the data set up to 35% of missing values: less than 5% of imputed data points are clustered to other clusters in comparison with the cluster distribution of the complete data set, and the cluster centers are approximately the same.

Figure 6 presents the RMSE, MAE, UCE, and CCD for the data set D500 with no structure. From here we conclude that if data do not have well separated clusters or a clear structure then the imputations produced by mean and regression are as good as those produced by the other two methods. In fact, the evaluation metrics show a little advantage to the mean and regression over the more sophisticated 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 causes failure in the prediction phase of the IVIACLR.

Figures 7 and 8 present the results for U10000 data set. Figure 7 includes all the imputation methods while Figure 8 compares only MICE and the IVIACLR. Here, similar to U500 data set, the IVIACLR produces the most accurate imputations due to the clear structure of the data set. Note that Figures 7(c) and 7(d) show the usefulness of the CCD metric: the sole UCE indicates that the mean is as good as MICE and IVIACLR up to 25% of missing values, however the CCD shows a large bias in the values imputed by the mean.

3) Performance of IVIACLR *on real-world data:* In this subsection, we demonstrate the performance of the IVIACLR in real life applications and compare it with other imputation methods using the real-world data given in Table II. These data sets are different from each other and our aim is to demonstrate the strengths/weaknesses of the IVIACLR. In addition to the MCAR data, we test the performance of the methods under the MAR and MNAR missing data mechanisms with Iris and Wine recognition data. The MAR and MNAR data used in our experiments are available at https://github.com/SnTa2019/Missing-Value-Imputation.

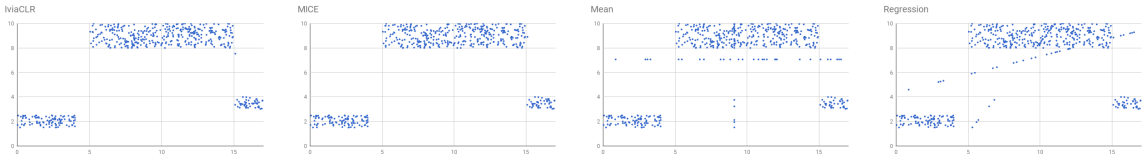
Iris plant is a well-known and widely used data set in testing data mining tools. It contains three clusters — one well separated and two overlapping. In this data the IVIACLR produces accurate imputations even when 45% of the data is missing (see Figure 9). The accuracies of these imputations are similar or even better than those with MICE. Furthermore, both the IVIACLR and MICE produce clearly more accurate imputations than the mean and regression. This confirms the conclusion obtained with artificial data that the IVIACLR produces very accurate imputations in data with a clear cluster structure. The evaluation metrics obtained with IVIACLR are similar regardless of the missing value mechanism when only 5% of the data is missing (see Figures 10 and 11). With larger portions of missing values there can be seen differences in the UCEs: e.g. with 25% of values missing, only 3% of points are misclassified in MCAR data after imputation while with MNAR data this error is 8%.

Wine recognition is a good data set for preliminary testing of data mining tools, yet it is not very challenging. It has three classes with "well-behaved" class structures. The number of features in this data is larger (13) than that of Iris plant (4). Note that when the number of features or the percentage of missing values in data is a bit larger the regression gives very inaccurate imputations (i.e. very large errors and CCDs, see [62]). This is due to the fact that the complete data set, obtained by deleting all incomplete objects and used for regression, becomes very small (or it does not exist). To make the figures more illustrative, we omit the regression imputation with Wine recognition, Red wine, and Abalone data sets. With Wine recognition data all the imputation methods but regression, produce imputations with UCE = 0, with all percentages of missing values and all missing data mechanisms (see Figures 12–14). This is due to the existence of the very clear cluster structure of the data. However, here we see differences in CCDs between missing data mechanisms. This means that although the points are clustered similarly to the original data, the cluster centers are changed more in MNAR than in MCAR data indicating bias in imputed values in MNAR data. Nevertheless, the bias with the IVIACLR is smaller than that with the other methods tested. Again, we conclude that the IVIACLR outcomes with accurate imputations when there is a clear cluster structure in data.

The (very limited) test results with different missing data mechanisms show that there may be a small perturbation in the imputation obtained with the IVIACLR in MNAR data. In MAR data the produced imputations are almost as accurate as in MCAR data. Thus, we can say that the proposed method is well suited for MCAR and MAR data but in data with MNAR missing data mechanism the produced imputations can be accepted only with care. Nevertheless, the IVIACLR seems to produce more accurate imputations than e.g. the well-known MICE. This is probably due to the CLR procedure used to predict the missing values: CLR provides a good approximation of a data and a good prediction will be obtained assuming that the data set itself is representative and there are prototypes among the set of complete objects.

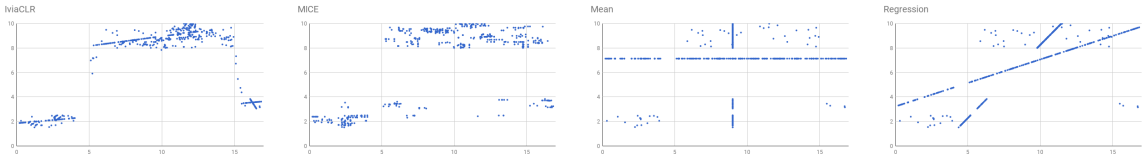
The data set TSLIB1060 has four small clear clusters and many noisy points distributed in the middle (see Figure 15). Figures 16 and 17 show that imputations produced by the IVIACLR and MICE preserve this structure to some extent even with 45% of values missing, while the mean and regression tend to impute all the values to the middle area of the data space. This is in accordance with the fact that the CCD shows very large bias in centers of clusters with the mean and regression imputations for larger percentages of missing data (see Figure 18). For all methods the number of misclassified data points is large (UCE $>$ 40%) when there are more than 25% of missing values. With the percentage of missing values less than 25% the IVIACLR performs as good as MICE according to the UCE and CCD, and it always has smaller errors RMSE and MAE.

Fig. 2. Original U500 with no missing values



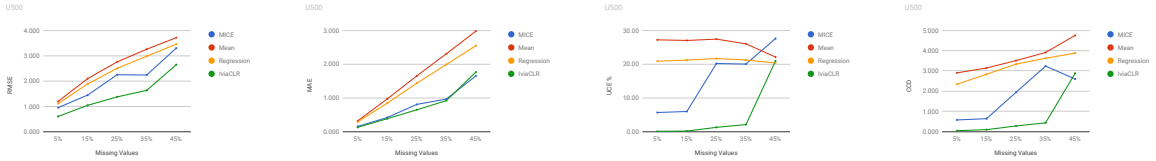
(a) Imputation by IVIACLR (b) Imputation by MICE (c) Imputation by Mean (d) Imputation by Regression

Fig. 3. U500: imputed data sets with 5% of MCAR data.



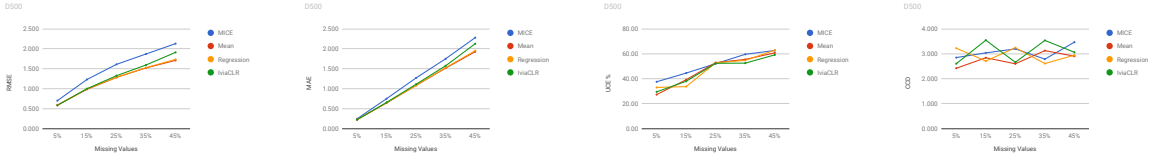
(a) Imputation by IVIACLR (b) Imputation by MICE (c) Imputation by Mean (d) Imputation by Regression

Fig. 4. U500: imputed data sets with 45% of MCAR data.



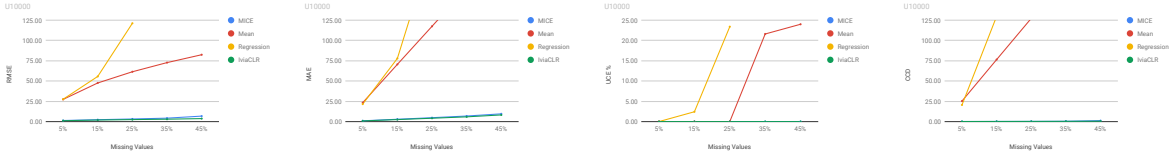
(a) RMSE (b) MAE (c) UCE (d) CCD

Fig. 5. U500: RMSE, MAE, UCE, and CCD versus the number of MCAR values.



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

Fig. 6. D500: RMSE, MAE, UCE, and CCD versus the number of MCAR values.

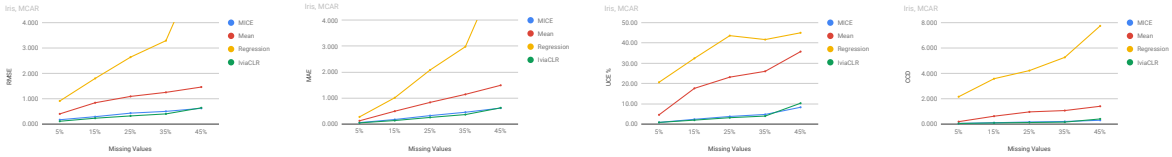


(a) RMSE (b) MAE (c) UCE (d) CCD

Fig. 7. U10000: RMSE, MAE, UCE, and CCD versus the number of MCAR values.

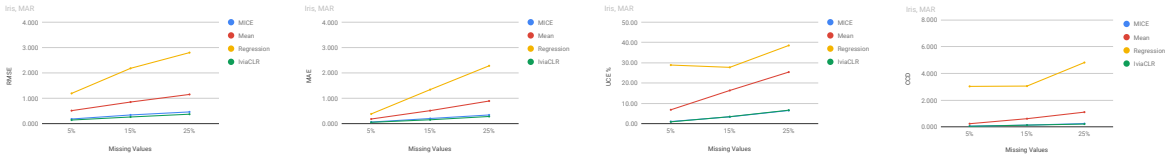
(a) RMSE (b) MAE (c) UCE (d) CCD

Fig. 8. U10000: RMSE, MAE, UCE, and CCD versus the number of MCAR values for MICE and IVIACLR.



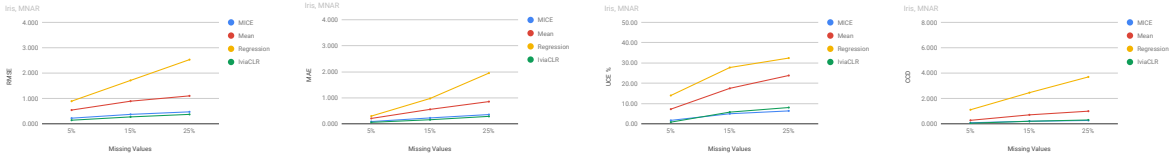
(a) RMSE (b) MAE (c) UCE (d) CCD

Fig. 9. MCAR Iris: RMSE, MAE, UCE, and CCD versus the number of missing values.



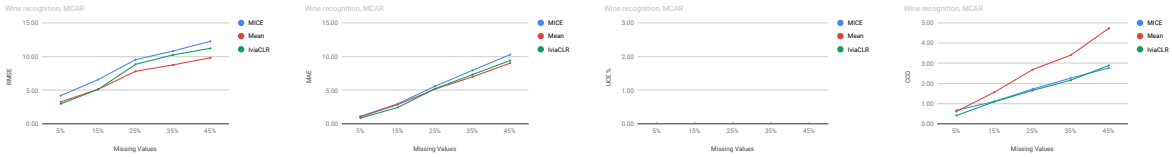
(a) RMSE (b) MAE (c) UCE (d) CCD

Fig. 10. MAR Iris: RMSE, MAE, UCE, and CCD versus the number of missing values.



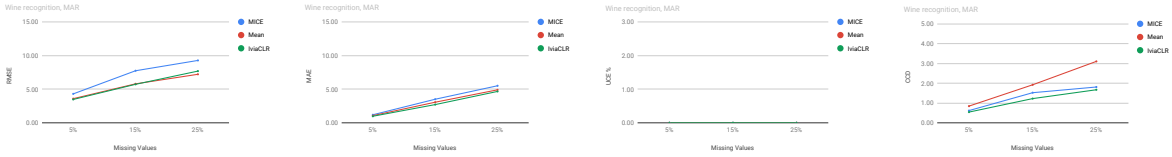
(a) RMSE (b) MAE (c) UCE (d) CCD

Fig. 11. MNAR Iris: RMSE, MAE, UCE, and CCD versus the number of missing values.



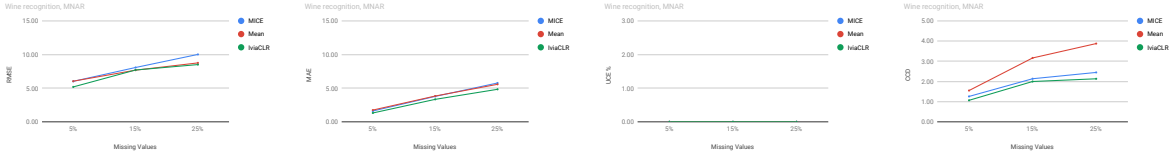
(a) RMSE (b) MAE (c) UCE (d) CCD

Fig. 12. MCAR wine recognition: RMSE, MAE, UCE, and CCD versus the number of missing values.



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

Fig. 13. MAR wine recognition: RMSE, MAE, UCE, and CCD versus the number of missing values.



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

Fig. 14. MNAR wine recognition: RMSE, MAE, UCE, and CCD versus the number of missing values.

In Red wine quality data all the samples can be approximated exactly using 6 hyperplanes. However, the number of samples in each hyperplane is not balanced (this is because there are much more normal wines than excellent or poor ones). Here, the IVIACLR is the best imputation method according to all evaluation metrics with small amount of missing values ($\leq$ 15%, see Figure 19). However, when more values are missing both the UCE and CCD indicate that the structure of the data is lost. This is due to the fact that although the underlying CLR procedure in the IVIACLR models the data correctly, the prediction phase fails with large number of missing values and, therefore, new values are imputed (at least near) to those hyperplanes with most samples. Nonetheless, the percentages of misclassified data points are large with all the methods when there are more than 25% of missing values.

Abalone has 29 strongly overlapping classes. Nevertheless, we use only two regression functions to model this data (similar to the number of clusters used in [30]). Here, the IVIACLR produces imputations with the smallest errors but both the mean and MICE preserve the original structure of the data better (see Figure 20). These results match with those of artificial data set D500 with no structure.

VI. Conclusions and Discussion

A new method IVIACLR (imputation via clusterwise linear regression) for imputing missing values of incomplete data was proposed in this paper. The approach is based on clusterwise linear regression (CLR), and it simultaneously finds optimal clusters within the data and their associated regression functions. CLR provides a good approximation of a data and accurate predictions for missing values will be obtained assuming that the data set itself is representative and there are prototypes among the set of complete objects. In other words, the idea is to approximate missing values using only those data points which are somewhat similar to the incomplete data object. In addition, we introduced a new cluster center displacement (CCD) metrics 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 CCD. The results confirm that the IVIACLR usually produces values that lead to smaller errors (RMSE and MAE) than the well-known imputation method MICE. In addition, the UCE and CCD indicate that the IVIACLR preserves well the original structure of the data set with small and moderate percentages of missing values ($\leq$25%). With larger percentages of missing data MICE usually computes more accurate imputation in terms of the UCE and CCD, but not necessary in terms of RMSE and MAE.

In the current version of the 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). In addition, although the IVIACLR is used here as a single imputation method, it could be used as a multiple imputation (MI) method (comparably e.g. with MICE). This can be done by taking into account all the intermediate results obtained during the CLR process, by considering predictions provided by different regression functions as multiple imputations, using various prediction methods, and/or using different initial imputations.

We conclude that the IVIACLR produces the most accurate imputations in MCAR and MAR data sets with a clear structure and small or moderate percentages of missing values. In these cases, it is the most accurate imputation method tested. Nevertheless, the statistical analysis of the IVIACLR in imputation of right, left and interval- censored data [63] is considered as a future work.

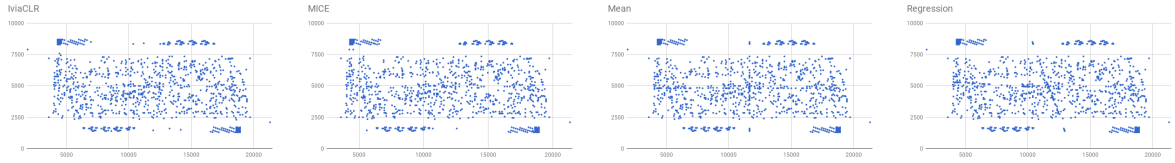
Acknowledgment

We would like to thank two anonymous reviewers for their valuable comments. The work was financially supported by the Academy of Finland (Project No. 289500, 294002, and 313269) and by the Australian Goverment through the Australian Research Counsil's Discovery Projects funding scheme (Project No. DP190100580).

References

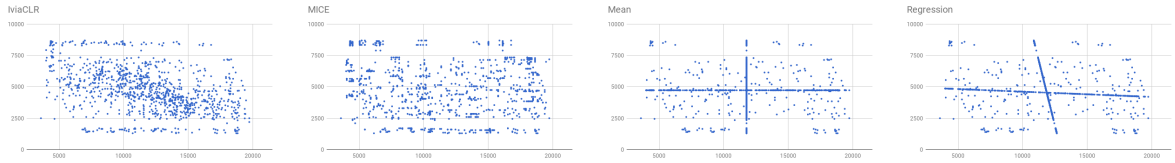
[1] E. Hazan, R. Livni, Y. Mansour, Classification with low rank and missing data, in: Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 2015.
[2] Shin-mu Tseng, Kuo-ho Wang, Chien-i. Lee, A pre-processing method to deal with missing values by integrating clustering and regression techniques, Applied Artificial Intelligence 17 (5–6) (2003) 535–544.
[3] L. Yao, K. Weng, Imputation of incomplete data using adaptive ellipsoids with linear regression, Journal of Intelligent & Fuzzy Systems 29 (2015) 253–265.
[4] C. Enders, Applied Missing Data Analysis, The Guilford Press, 2010.
[5] R. Little, D. Rubin, Statistical Analysis with Missing Data, 2nd Edition, John Wiley and Sons, 2002.
[6] J. T. Chi, E. C. Chi, R. G. Baraniuk, k-POD: A method for k-means clustering of missing data, The American Statistician 70 (2016) 91–99.
[7] H. Liu, M. Shao, Z. Ding, Y. Fu, Structure-preserved unsupervised domain adaptation, IEEE Transactions on Knowledge and Data Engineering 31 (4) (2018) 799–812.
[8] K. Wagstaff, Clustering with missing values: No imputation required, in: D. Banks, F. McMorris, P. Arabie, W. Gaul (Eds.), Classification, Clustering, and Data Mining Applications. Proceedings of the Meeting of the International Federation of Classification Societies (IFCS), Illinois Institute of Technology, Chicago, Springer, 2004, pp. 649–658.
[9] J. Wu, H. Liu, H. Xiong, J. Cao, J. Chen, K-Means-based consensus clustering: A unified view, IEEE Transactions on Knowledge and Data Engineering 27 (1) (2015) 155–169.
[10] R. Hathaway, J. Bezdek, Fuzzy c-means clustering of incomplete data, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 31 (1) (2001) 735 – 744.
[11] L. Zhang, W. Lu, X. Liu, W. Pedrycz, C. Zhong, Fuzzy c-means clustering of incomplete data based on probabilistic information granules of missing values, Knowledge-Based Systems 99 (C) (2016) 51–70.
[12] B. Gabrys, Neuro-fuzzy approach to processing inputs with missing values in pattern recognition problems, International Journal of Approximate Reasoning 30 (3) (2002) 149–179.
[13] H. Gao, X.-W. Liu, Y.-X. Peng, S.-L. Jian, Sample-based extreme learning machine with missing data, Mathematical Problems in Engineering 2015 (2015) 1–11.
[14] K. Pelckmans, J. D. Brabanter, J. A. K. Suykens, B. D. Moor, Handling missing values in support vector machines classifiers, Neural networks 18 (5–6) (2005) 684–692.
[15] P. K. Sharpe, R. J. Solly, Dealing with missing values in neural network-based diagnostic systems, Neural Computing and Applications 3 (2) (1995) 73–77.
[16] Y. Yan, Y. Zhang, J. Chen, Y. Zhang, Incomplete data classification with voting based extreme learning machine, Journal Neurocomputing 193 (C) (2016) 167–175.
[17] M. Hejazi, S. A. R. Al-Haddad, Y. P. Singh, A. F. Aziz, Multiclass support vector machines for classification of ECG data with missing values, Applied Artificial Intelligence 29 (7) (2015) 660–674.
[18] M. Mojirsheibani, T. Reese, Kernel regression estimation for incomplete data with applications, Statistical Papers 58 (1) (2015) 185–209.

Fig. 15. Original TSPLIB1060 with no missing values



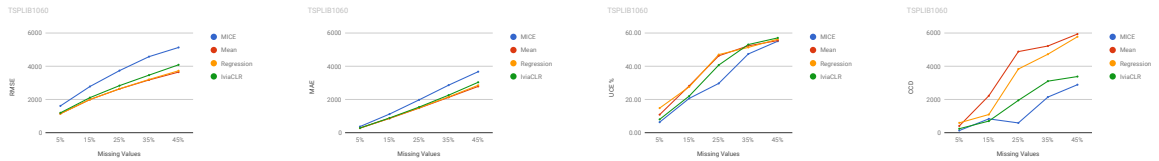
(a) Imputation by IVIACLR (b) Imputation by MICE (c) Imputation by Mean (d) Imputation by Regression

Fig. 16. TSPLIB1060: imputed data sets with 5% of MCAR data.



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

Fig. 17. TSPLIB1060: imputed data sets with 45% of MCAR data.



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

Fig. 18. TSPLIB1060: RMSE, MAE, UCE, and CCD versus the number of MCAR values.

[19] W. Tang, H. He, D. Gunzler, Kernel smoothing density estimation when group membership is subject to missing, Journal of Statistical Planning and Inference 142 (3) (2012) 685–694.

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

[21] T. Raghunathan, J. Lepkowski, J. van Hoewyk, P. Solenberger, A multivariate technique for multiply imputing missing values using a sequence of regression models, Survey Methodology 27 (1) (2001) 85–95.

[22] Y. Qin, S. Zhang, X. Zhu, J. Zhang, C. Zhang, Semi-parametric optimization for missing data imputation, Applied Intelligence 27 (2007) 79–88.

[23] I. White, P. Royston, A. Wood, Multiple imputation using chained equations: Issues and guidance for practice, Statistics in Medicine 30 (4) (2011) 377–399.

[24] Q. H. Wang, R. Rao, Empirical likelihood-based inference under imputation for missing response data, Annals of Statistics 30 (2002) 896–924.

[25] D. Williams, X. Liao, Y. Xue, L. Carin, B. Krishnapuram, On classification with incomplete data, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (3) (2007) 427–436.

[26] A. Gupta, M. S. Lam, Estimating missing values using neural networks, Journal of the Operational Research Society 47 (2) (1996) 229–238.

[27] E.-L. Silva-Ramíreza, R. Pino-Mejíasb, M. López-Coelloa, M.-D. Cubiles-de-la Vegac, Missing value imputation on missing completely at random data using multilayer perceptrons, Neural Networks 24 (1) (2011) 121–129.

[28] D. Li, J. Deogun, W. Spaulding, B. Shuart, Towards missing data imputation: A study of fuzzy k-means clustering method, in: S. Tsumoto, R. Słowiński, J. Komorowski, J. Grzymała-Busse (Eds.), Rough Sets and Current Trends in Computing. 4th International Conference, RSCTC 2004, Uppsala, Sweden., Springer, 2004, pp. 573–579.

[29] B. M. Patil, R. C. Joshi, D. Toshniwal, Missing value imputation based on k-mean clustering with weighted distance., in: R. S. et al. (Ed.), Contemporary Computing. IC3 2010. Communications in Computer and Information Science, vol 94, Springer, Berlin, Heidelberg, 2010, pp. 600–609.

[30] C. C Zhang, Y. Qin, X. Zhu, J. Zhang, Z. S. Clustering-based missing value imputation for data preprocessing, Industrial Informatics, 2006 IEEE International Conference (2006).

[31] X. Zhu, S. Zhang, Z. Jin, Z. Zhang, Z. Xu, Missing value estimation for mixed-attribute data sets, IEEE Transactions on Knowledge and Data Engineering 23 (1) (2011) 110–121.

[32] G. Batista, M. C. Monard, A study of k-nearest neighbor as an imputation method, in: Abraham, A. et. al. (Ed.), Hybrid Intelligent Systems, IOS Press, 2002, pp. 251–260.

[33] G. Tutz, S. Ramzan, Improved methods for the imputation of missing data by nearest neighbor methods, Computational Statistics and Data Analysis 90 (2015) 84–99.

[34] F. H. Clarke, Optimization and Nonsmooth Analysis, Wiley-Interscience, New York, 1983.

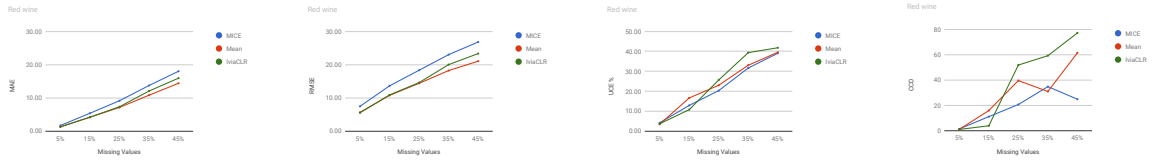
[35] A. M. Bagirov, N. Karmitsa, M. M. Mäkelä, Introduction to Nonsmooth Optimization: Theory, Practice and Software, Springer, 2014.

[36] A. M. Bagirov, J. Ugon, H. Mirzayeva, Nonsmooth nonconvex optimization approach to clusterwise linear regression problems, European Journal of Operational Research 229 (1) (2013) 132–142.

[37] A. M. Bagirov, J. Ugon, H. Mirzayeva, An algorithm for clusterwise linear regression based on smoothing techniques, Optimization Letters 9 (2) (2015) 375–390.

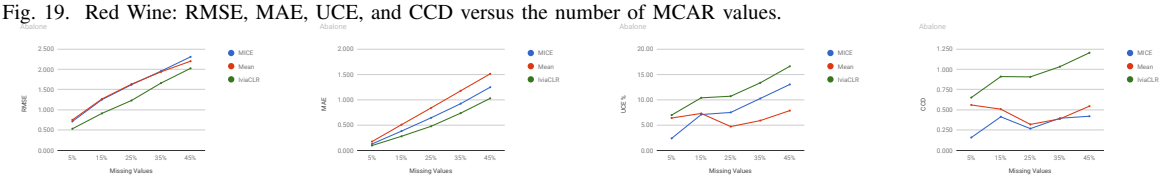
[38] A. M. Bagirov, J. Ugon, H. G. Mirzayeva, Nonsmooth optimization algorithm for solving clusterwise linear regression problem, Journal of Optimization Theory and Applications 164 (2015) 755–780.

[39] A. M. Bagirov, A. Mahmood, A. Barton, Prediction of monthly rainfall in victoria, australia: Clusterwise linear regression approach, Atmospheric Research 188 (2017) 20–29.



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

Fig. 19. Red Wine: RMSE, MAE, UCE, and CCD versus the number of MCAR values.



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

Fig. 20. Abalone: RMSE, MAE, UCE, and CCD versus the number of MCAR values.

[40] L. He, G. H. Huang, H. W. Lu, Health-risk-based groundwater remediation system optimization through clusterwise linear regression, Environmental Science & Technology 42 (24) (2008) 9237–9243.

[41] Y. Park, Y. Jiang, D. Klabjan, L. Williams, Algorithms for generalized clusterwise linear regression, INFORMS Journal on Computing 29 (2) (2017) 301–317.

[42] J.-M. Poggi, B. Portier, PM10 forecasting using clusterwise regression, Atmospheric Environment 45 (38) (2011) 7005–7014.

[43] C. Preda, G. Saporta, Clusterwise pls regression on a stochastic process, Computational Statistics & Data Analysis 49 (2005) 99–108.

[44] M. Wedel, C. Kistemaker, Consumer benefit segmentation using clusterwise linear regression, International Journal of Research in Marketing 6 (1) (1989) 45–59.

[45] S. Gaffney, P. Smyth, Trajectory clustering using mixtures of regression models, in: S. Chaudhuri, D. Madigan (Eds.), Proceedings of the ACM Conference on Knowledge Discovery and Data Mining, New York, 1999, pp. 63–72.

[46] H. Späth, Algorithm 39: Clusterwise linear regression, Computing 22 (1979) 367–373.

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

[48] W. DeSarbo, W. Cron, A maximum likelihood methodology for clusterwise linear regression, Journal of Classification 5 (2) (1988) 249–282.

[49] L. Garcìa-Escudero, A. Gordaliza, A. Mayo-Iscar, R. San Martin, Robust clusterwise linear regression through trimming, Computational Statistics and Data Analysis 54 (2010) 3057–3069.

[50] W. DeSarbo, R. Oliver, A. Rangaswamy, A simulated annealing methodology for clusterwise linear regression, Psychometrika 54 (4) (1989) 707–736.

[51] R. Carbonneau, G. Caporossi, P. Hansen, Globally optimal clusterwise regression by mixed logical-quadratic programming, European Journal of Operational Research 212 (2011) 213–222.

[52] R. Carbonneau, G. Caporossi, P. Hansen, Extensions to the repetitive branch-and-bound algorithm for globally-optimal clusterwise regression, Computers and Operations Research 39 (11) (2012) 2748–2762.

[53] N. Karmitsa, A. Bagirov, S. Taheri, K. Joki, Limited memory bundle method for clusterwise linear regression, in: T. Tuovinen, J. Periaux, P. Neittaanmäki (Eds.), Computational Sciences and Artificial Intelligence in Industry, Springer, 2020, in-press.

[54] M. Haarala, K. Miettinen, M. M. Mäkelä, New limited memory bundle method for large-scale nonsmooth optimization, Optimization Methods and Software 19 (6) (2004) 673–692.

[55] N. Haarala, K. Miettinen, M. M. Mäkelä, Globally convergent limited memory bundle method for large-scale nonsmooth optimization, Mathematical Programming 109 (1) (2007) 181–205.

[56] M. Azur, E. Stuart, C. Frangakis, P. Leaf, 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.

[57] S. van Buuren, MICE, Available online at <URL: https://cran.r-project.org/web/packages/mice/mice.pdf> (February 28th, 2018).

[58] R. M. Schouten, P. Lugtig, G. Vink, Generating missing values for simulation purposes: a multivariate amputation procedure, Journal of Statistical Computation and Simulation 88 (15) (2018) 2909–2930.

[59] R. Schouten, P. Lugtig, J. Brand, G. Vink, Generate missing values with ampute, Available online at <URL: https://rianneschouten.github.io/mice_ampute/vignette/ampute.html> (January 28th, 2019) (2018).

[60] D. Dua, E. Karra Taniskidou, UCI machine learning repository, Available online at <URL: http://archive.ics.uci.edu/ml>, University of California, Irvine, School of Information and Computer Sciences (2017).

[61] N. Karmitsa, A. Bagirov, S. Taheri, Clustering in large data sets with the limited memory bundle method, Pattern Recognition 83 (2018) 245–259.

[62] N. Karmitsa, S. Taheri, A. Bagirov, P. Mäkinen, Clusterwise linear regression based missing value imputation for data preprocessing, TUCS Technical Report, No. 1193, Turku Centre for Computer Science, Turku, the report is available online at <URL: http://napsu.karmitsa.fi/publications/iviaclr_tucs.pdf> (2018).

[63] Y. V. Karpievitch, A. R. Dabney, R. D. Smith, Normalization and missing value imputation for label-free LC-MS analysis, BMC Bioinformatics 13 (S5).

Napsu Karmitsa (nee Haarala) received her Ph.D. degree from the University of Jyvaskyla, Finland, in 2004. At the moment, she holds a position of an Academy Research Fellow granted by the Academy of Finland. In addition, she is an Adjunct Professor in applied mathematics at the University of Turku, Finland. Karmitsa's research is focused on nonsmooth optimization and analysis. Special emphasis is given to nonconvex, global and large-scale cases, and applications in data mining.

Sona Taheri received her Ph.D. degree from the University of Ballarat, Australia, in 2012. She is now a Research Fellow at Federation University Australia, Ballarat, Australia. Her current research interests include numerical optimization and data mining.

Adil Bagirov received his Ph.D. degree from the University of Ballarat, Australia, in 2002. He is now an Associate Professor at Federation University Australia, Ballarat, Australia. His current research interests include numerical optimization and data mining.

Pauliina Mäkinen received her MSc. degree from the University of Turku, Finland, in 2018. She is now a Postgraduate Student at Department of Mathematics and Statistics, and Future Technologies at University of Turku. Her research interests include numerical optimization, data mining, and machine learning.