Expert Systems With Applications 204 (2022) 117298

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

# A bi-objective $k$-nearest-neighbors-based imputation method for multilevel data

Maximiliano Cubillos [*], Sanne Wøhlk, Jesper N. Wulff

*Econometrics and Business Analytics, Department of Economics and Business Economics, Aarhus University, Fuglesangs Allé 4, DK-8210 Aarhus V, Denmark*

ARTICLE INFO

*Keywords:*
Multilevel data
Imputation
$k$-nearest neighbors

ABSTRACT

We propose a bi-objective algorithm based on the $k$-nearest neighbors (biokNN) method to perform imputation of missing values for data with multilevel structures with continuous variables. We define the imputation method as a bi-objective minimization problem and propose a solution algorithm based on a weighted objective function. The algorithm seeks imputed values that balance the dissimilarity between the $k$-nearest neighbors and the observations within the same cluster. The effectiveness of the proposed method is evaluated through a simulation study, and its results are compared with those of eight benchmark imputation methods. The simulation study is based on the generation of datasets with a varying-intercept–varying-slope multilevel model, and the results are compared both by using well-known accuracy metrics and by estimating the bias of the estimates after inference has been performed. Based on the simulation, the effects of different configurations of multilevel datasets are tested, including the number of clusters, their size, their similarity, the percentage of missing values, and the effect of imbalanced clusters. The results show that the proposed method outperforms the benchmark methods, especially in cases with high intraclass correlation. A comparison of fitted linear multilevel regression models shows that our method can also reduce the bias of the estimates and the coefficient of determination. Finally, the method is tested on three commonly used machine learning datasets and shows better accuracy in most cases compared with the benchmark methods.

## 1. Introduction

The presence of incomplete data is a common problem in most intelligent systems applications, limiting the implementation and analysis of statistical and machine learning models (Lin & Tsai, 2020). To minimize the loss of efficiency and the bias that arise from removing rows with missing data, statisticians recommend the use of imputation algorithms (Horton & Kleinman, 2007). These algorithms use the observed data to estimate values that can replace the missing values (Garciarena & Santana, 2017). However, for imputation algorithms to work optimally on clustered data, they need to account for cluster effects in the data (Andridge, 2011; Drechsler, 2015; Goldstein et al., 2014). When datasets show some form of natural clustering where lower-level units (e.g., students or employees) are nested with higher-level units (e.g., classrooms or departments), they are said to have a multilevel structure. Such multilevel structure needs to be accounted for by the imputation algorithm, since ignoring it may result in severe model and parameter misspecification (Black et al., 2011; Enders et al., 2016).

One of the most widely used approaches for handling multilevel data structures with incomplete data is multiple imputation using multilevel modeling (Grund et al., 2018). In multiple imputation, several copies of the dataset are generated, each with different plausible replacement values, and then used in a second phase to perform analysis by pooling the results (Enders et al., 2016). One of the advantages of a multilevel approach to missing data is that it can model the intraclass correlation among levels directly, thereby providing more accurate estimates of each cluster and thus reducing overfitting (Carpenter & Kenward, 2012). This approach, however, can be vulnerable to distributional and model misspecification (Grund et al., 2016). In addition, according to Grund et al. (2018), “the imputation model must be at least as general as the analysis model”. This makes it hard for the analyst to remain flexible, even though they are not sure a priori which kind of multilevel model they wish to estimate. To cope with this limitation, in this paper, we propose a new method for imputation that is indeed both simple and generic, and that can outperform state-of-the-art methods for multilevel imputation in most scenarios.

We develop a novel bi-objective imputation method based on the $k$-nearest neighbors (kNN) algorithm. Bi-objective kNN imputation, or biokNN for short, minimizes the distance between the imputed values,

[*] Corresponding author.
*E-mail addresses:* mcub@econ.au.dk (M. Cubillos), sanw@econ.au.dk (S. Wøhlk), jwulff@econ.au.dk (J.N. Wulff).

https://doi.org/10.1016/j.eswa.2022.117298
Received 24 May 2021; Received in revised form 8 February 2022; Accepted 22 April 2022
Available online 16 May 2022

their neighbors, and class neighbors by solving a bi-objective optimization problem. It does not require any distributional assumptions or model specification. This flexibility is a major advantage, especially when the user wishes to use several different multilevel analysis models directed at different research questions (Grund et al., 2018). Using a comprehensive simulation study, we demonstrate that biokNN outperforms the current state-of-the-art imputation algorithms for multilevel data in most cases. It is particularly superior when missing data rates are high and clusters are unbalanced. We confirm the performance of biokNN over alternative algorithms on three benchmark datasets.

The contribution of this paper is twofold. First, we present a new imputation method designed for datasets with continuous variables having a multilevel structure. This method is based on the kNN method and is formulated as a bi-objective optimization problem. Second, we present an extensive simulation study that tested different configurations of multilevel models and compared the imputation accuracy and inference performance with those of eight other imputation methods. In general, we find that biokNN gives better imputation accuracy in most of the scenarios.

The rest of this paper is structured as follows. Section 2 presents the most appropriate imputation methods and their applications to multilevel structured data. Section 3 presents the methodology, including the formulation of the imputation problem as an optimization problem, the proposed biokNN method, the simulation setup, the benchmark imputation methods used for comparison, and the comparison metrics. In Section 4, the results of both the simulation study and the applications to the benchmark datasets are presented. This is followed by a discussion of the implications of the results, the limitations of the method, and directions for future work are presented in Section 5. Finally, Section 6 concludes the paper.

## 2. Literature review

The problem of missing data has been studied extensively in statistics, given its broad applicability in many fields of research. There are two main approaches to dealing with missing values, namely, deletion and imputation (Sefidian & Daneshpour, 2019). Deletion methods ignore cases or variables in which there are missing values, and, owing to their simplicity, can be useful in cases with low rates of missing values (Lan et al., 2020). However, when the rate of missing values is high, deletion can cause a major loss of information and lead to bias and overfitting in the resulting models (Purwar & Singh, 2015). In such cases, imputation methods are preferred, with the observed information being used to estimate the missing values.

The most straightforward imputation method is mean imputation, in which a missing value is replaced by the mean of the observed values of the variable. This imputation method is simple, but it usually results in poor imputation accuracy since it ignores the correlation between the variables in the dataset (Little & Rubin, 2019). Regression imputation incorporates the correlation of the variables in the dataset by replacing a missing value with the least squares estimate of its regression on all of the other variables in the data (Raghunathan et al., 2001). By contrast, the predictive-mean matching method imputes the missing values by drawing random samples from a set of observed values close to regression predictions (Groothuis-Oudshoorn & Van Buuren, 2011). These methods have been extended to include both numerical and categorical variables and to use other estimation methods such as support vector regression (Lin & Tsai, 2020). One of the main drawbacks of imputation methods based on regression is that they have to meet strong assumptions and may perform poorly on datasets with nonlinear relationships between variables. Recently, methods based on state-of-the-art machine learning techniques have proven to be useful for imputation purposes. Song and Sun (2020) study the distance models that predict distances between tuples for missing data imputation using distance likelihood maximization. Awawdeh et al. (2022) propose an imputation method that performs feature selection simultaneously to enhance the learning performance of the model using an evolutionary approach. Finally, Lin et al. (2022) compare multilayer perception and deep belief networks for missing value imputation and propose two differently ordered combinations of data discretization.

One of the most widely used nonparametric imputation methods is kNN imputation. In this method, a missing value from a given variable is replaced by the mean of the $k$-nearest neighbors of the observations from the same variable. Different distance functions can be used to select the neighbors, which allows the method to include both numerical and categorical variables. A main advantage of kNN is that it does not need specification of any predictive model. Furthermore, it has a simple implementation, and it usually provides good performance compared with other methods (Jiang & Yang, 2015). Troyanskaya et al. (2001) compared kNN imputation with mean imputation and singular-value decomposition (SVD) techniques. Based on simulations, their study showed that kNN performs well compared with mean and SVD imputation.

Several imputation methods based on kNN have been proposed in the literature, and its effectiveness compared with other imputation methods has been demonstrated. Caruana (2001) presented an iterative kNN method that refines the imputed values and chooses the nearest neighbors based on the estimated values from the previous iteration. Kim et al. (2004) proposed a sequential kNN imputation method that starts by imputing missing values from observations with the fewest missing dimensions, reuses the previously imputed values, and continues imputing the subsequent missing values. Kim et al. (2005) proposed a local least squares method based on kNN that imputes values using regression models trained on the nearest neighbors of a given observation. Variations of the local least squares kNN-based method using iterative and sequential methods were proposed by Cai et al. (2006) and Zhang et al. (2008), respectively. Tutz and Ramzan (2015) proposed a weighted nearest neighbor imputation method that uses distances for selected variables as weights in the imputation process. The weight of the imputed values is assigned individually for each observation, in contrast to the weighted approach used in this paper, in which the weight is assigned in the objective function. In a similar fashion, Pan et al. (2015) proposed a method that uses mutual information weighted gray relational analysis to obtain the similarity metric in the kNN method and thereby determine the nearest neighbors of a missing observation. Similar to Kim et al. (2005), Rachdi et al. (2021) presented a method combining the kNN method with a local linear estimation approach when the regressor is of functional type and the response variable is numerical but observed with some missing at random observations. Finally, Al-Helali et al. (2021) combined a weighted kNN method with genetic programming, with kNN being used to select instances to construct the genetic models. None of these previous variations of the kNN method, however, have explored the integration of the multilevel structure into the model with a bi-objective function, as is done in this study.

Imputation methods that address the multilevel structure of the data are usually based on linear regression models with fixed or varying intercepts for the classes (Drechsler, 2015). There are two main procedures to integrate this multilevel structure into the model, namely, the joint modeling (JM) approach and the fully conditional specification (FCS) approach (Grund et al., 2018). In the JM approach, a single model is specified for all variables with missing data, while in the FCS model, missing data are imputed separately for each variable (Carpenter & Kenward, 2012). These two models have been extensively studied in recent years, but there are still limitations to their application, particularly when the sample size is limited or there are multiple interactions. Also, it has been argued that the two approaches imply similar structures and can be used interchangeably in some situations (Lüdtke et al., 2017; Mistler, 2015). Moreover, JM and FCS methods are usually based on multiple imputation, which aims to estimate the posterior distribution of the missing variables and the correlation between them and the other variables present in the dataset (Tutz & Ramzan, 2015). This makes the two approaches computationally more expensive compared with nonparametric methods, since they require Markov chain Monte Carlo estimations (Drechsler, 2015).

## 3. Methodology

In this section, we present the methodology used in this study. In Section 3.1, the problem description is presented. This includes the formulation of the multilevel imputation optimization problem. Section 3.2 describes the algorithm used to solve the proposed bi-objective imputation problem. Then, Section 3.3 presents the details of the simulation study used to assess the performance of the proposed method. This subsection describes the multilevel varying slope model, and summarizes the parameters used in the simulation.

### 3.1. Problem description

This subsection describes the formulation of the missing value imputation optimization problem for continuous variables, together with the proposed solution method using a bi-objective kNN-based algorithm. The optimization problem can be described as follows. Let $X \in \mathbb{R}^{n\times(P+1)}$ be a dataset with $P$ continuous variables, $n$ observations, and one class variable $X_q$ containing $Q$ classes. The observations in the dataset can be divided into two sets: the missing indexes $\mathcal{M} = \{(i, p)$, in which the value $x_{ip}$ is missing$\}$, and the indexes of the observed values $\mathcal{O} = \{(i, p)$, in which the value $x_{ip}$ is observed$\}$. We use an auxiliary set $\mathcal{I}$ as the set of indexes $i$ in which an observation has at least one missing value.

The objective of the proposed approach is to minimize the dissimilarity between the imputed values and both (1) its $k$-nearest neighbors and (2) its class neighbors. Since the two objectives can be conflictive, there is no single optimal solution for this problem. Instead, for bi-objective problems, sets of efficient solutions that represent the trade-off between the two objectives are to be found. In our application, we seek for a single solution since we require a single imputed dataset as output. A well-known approximation method for problems with more than one objective that do no alter the structure of our problem is the weighting method (Zadeh, 1963). We use this approach to combine the two objectives into a single objective by adding a positive parameter $\alpha$, $0 \le \alpha \le 1$, which define a convex combination of the two objectives. With this approach, the weighting parameter is an input of the problem.

For a given observation $i \in \{1, \dots, n\}$, the $k$-nearest neighbors part of the objective are given by the set of $k$ observations with the smallest distance from $i$. The quadratic Euclidean distance between two observations $(i, i')$ is given by the squared difference among the $P$ variables in the dataset:

$$d_{ij} = \sum_{p=1}^{P} (x_{ip} - x_{jp})^2 \tag{1}$$

On the other hand, the class neighbors $Q$ of an observation $i$ are given by the set of observations that belong to the same class, i.e., that share the same value in the class variable $X_q$.

The minimization is based on two decision variables: $w_{ip}$, representing the imputed value of $(i, p) \in \mathcal{M}$, and an auxiliary variable $z_{ij}$, representing the neighbor assignment of the algorithm based on the distance defined in Eq. (1). This auxiliary variable is defined by

$$z_{ij} = \begin{cases} 1 & \text{if } x_j \text{ is among the } k\text{-nearest neighbors of } x_i \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The purpose is to solve the following optimization problem:

$$\begin{aligned} \min \quad & \alpha \left\{ \sum_{i\in\mathcal{I}} \sum_{j=1}^{n} z_{ij} \left[ \sum_{p=1}^{P} (w_{ip} - w_{jp})^2 \right] \right\} \\ & + (1-\alpha) \left\{ \sum_{i\in\mathcal{I}} \sum_{j\in Q} \left[ \sum_{p=1}^{P} (w_{ip} - w_{jp})^2 \right] \right\} \end{aligned} \tag{3}$$

$$\begin{aligned} \text{s.t.} \quad & w_{ip} = x_{ip}, \quad (i,p) \in \mathcal{O} \\ & \sum_{j=1}^{n} z_{ij} = k, \quad i \in \mathcal{M} \\ & z_{ii} = 0 \\ & z_{ij} \in \{0, 1\} \end{aligned} \tag{4}$$

The objective function in Eq. (3) is the weighted sum of the two objectives, namely, the $k$ nearest neighbors and the class neighbors. The purpose of this function is to include the information given by the class variable in the imputed value, depending on the proportion $\alpha$. Note that the method is equivalent to the kNN method if $\alpha = 1$ and $k < n$, to the class mean method if $\alpha = 0$, and to the overall mean method if $\alpha = 1$ and $k = n$. Finally, the constraints in Eq. (4) include the observed values and ensure that the auxiliary variable $z_{ij}$ is well defined.

### 3.2. Solution approach

The problem formulated in (3)–(4) is a nonconvex optimization problem with both binary and continuous variables, which makes it difficult to solve to optimality. For that reason, we implement a solution algorithm based on the first-order coordinate descent method (Wright, 2015). The derivation and details on the implementation for the imputation problem are presented in Bertsimas et al. (2017). In our version, we modify the objective of the problem to include the weighted sum of the two objectives in our problem. The first-order coordinate descent method is an iterative approximate method and finding an global minimum is not guaranteed (Bertsekas, 1999).

An overview of the algorithm is presented in Algorithm 1, and it can be described as follows. First, three input parameters are required: the number of neighbors $k$, the weighting parameter $\alpha$, and the number of iterations $N_{\text{iter}}$. As a starting point, the missing values $x_{ij}$ are imputed by randomly assigning a sample from the variable $j$, and this is set as the start solution $X_0$. Then, for each iteration, we proceed in two steps. First, the algorithm updates the neighbor assignment by computing the distance matrix between the observations with at least one missing value in the original dataset $X$ and the rest of the observations. For each observation in $\mathcal{I}$ the distances are sorted, and the $k$ observations with the smaller distance are selected. Second, the imputed values $w_{iv}$ are updated. In this step, each imputed value is updated individually using a weighted average between the neighbors' mean value $w_{\text{neigh}}$ and the class mean value $w_{\text{class}}$. The process is repeated $N_{\text{iter}}$ times.

A reviewer commented that our algorithm resembles that of the traditional median filter (Tukey, 1977). Median filter takes the median over a sliding window of fixed size (Arias-Castro & Donoho, 2009) and is used for noise removal in signal and image processing (Barner & Arce, 2003; Caselles et al., 2000; George et al., 2018). To apply median filtering to imputation of multilevel data one would need to decide how to deal with cluster heterogeneity and the window size. Our algorithm approaches the multilevel imputation problem by phrasing it in terms of a bi-objective function solved by the implementation of a first-order coordinate descent method. Without such an addition, median filter would not be useful the multilevel imputation problem.

### 3.3. Simulation framework

We consider a varying-intercept–varying-slope multilevel model to simulate datasets with a multilevel structure and to be able to control its variables explicitly. A varying-slope model considers a target variable $y_{ij}$ that depends linearly on an independent variable $X_{ij}$, and a class variable, where $i \in \{1, \dots, n\}$ is the $i$th observation and $j \in \{1, \dots, Q\}$ is the $j$th class. The class variable contains the assignment of

**Algorithm 1**

```
Input: $X \in \mathbb{R}^{n\times(P+1)}$ dataset with missing values at indexes $(i,p) \in \mathcal{M}$
Input parameters: $k$, $\alpha$, $N_{\text{iter}}$
$X_0 \in \mathbb{R}^{n\times(P+1)}$ initial dataset imputed using random samples
$X' \leftarrow X_0$
while iter $\leq N_{\text{iter}}$ do
    Update neighbors' assignment:
    for each $i \in \mathcal{I}$ do
        compute distance between $i$ and all observations in $X'$
        sort the computed distances
        select the $k$ observations with the smallest distances
    end for
    Update the imputation for each missing value $(i,p)$
    for each $(i,p) \in \mathcal{M}$ do
        compute the class mean value $w_{\text{class}}$
        compute the neighbors' mean value $w_{\text{neigh}}$
        assign $w_{iv} \leftarrow \alpha w_{\text{neigh}} + (1-\alpha) w_{\text{class}}$
    end for
end while
Output: $X' \in \mathbb{R}^{n\times(P+1)}$ dataset with imputed values
```

$Q$ classes, each with $Q_s$ observations. The model can be formulated as follows:

$$y_{ij} = \beta_{0j} + \beta_{1j} X_{ij} + \epsilon \tag{5}$$

$$\beta_{0j} \sim \mathcal{N}(\mu_0, \tau_0) \tag{6}$$

$$\beta_{1j} \sim \mathcal{N}(\mu_1, \tau_1) \tag{7}$$

$$\epsilon \sim \mathcal{N}(0, \sigma) \tag{8}$$

where $\tau_0$ and $\tau_1$ represent the random effects of the intercepts and slopes among classes, respectively, and $\sigma$ corresponds to the overall random error of the model. The parameters $\mu_0$ and $\mu_1$ represent the average effects of the intercepts and slopes, respectively. As a baseline for comparison, we consider $\mu_0 = \mu_1 = \tau_1 = \sigma = 1$, while the variances of the intercepts vary, depending on the intraclass correlation, which we consider a simulation parameter.

The intraclass correlation measures how strongly the observations in the same class resemble each other, and it can be derived as

$$I = \frac{\tau_0}{\tau_0 + \sigma} \tag{9}$$

The values of $I$ range from 0 to 1, with $I = 0$ when the observations in the same class do not share characteristics, and $I = 1$ when they are exactly the same. In our simulation, we consider four representative cases with $I = \{0.3, 0.5, 0.7, 0.9\}$.

One of the effects on the imputation accuracy that we want to measure is that of the presence of unbalanced datasets. Unbalanced datasets are those in which the number of observations per class is uneven, with some classes presenting a high number of observations while some only have a few. To account for this in the simulation, we consider that the number of observations per class is drawn from a normal distribution:

$$Q_s \sim \mathcal{N}(\mu_{\text{class}}, \sigma_{\text{class}}) \tag{10}$$

where $\mu_{\text{class}}$ is the average number of observations per class and $\sigma_{\text{class}}$ is its variance. The baseline case considers $\sigma_{\text{class}} = 0$, where all classes have the same number of observations.

Once the parameters to simulate the observed or complete dataset have been chosen, random observations are made to be missing for both the target and the independent variable. The missing values are generated by taking a missing percentage $M$ and assuming them to be missing completely at random (MCAR, see below). The process is repeated $S$ times. Table 1 summarizes all the parameters included in the simulation study.

**Table 1**
Description of the simulation parameters.

| Variable | Explanation |
|---|---|
| $S$ | Number of simulations |
| $Q$ | Number of classes |
| $Q_s$ | Observations per class |
| $I$ | Intraclass correlation |
| $M$ | Percentage of missing values |
| $p$ | Number of variables |
| $\mu_0$ | Overall intercept mean |
| $\mu_1$ | Slope of variable $X$ mean |
| $\tau_0$ | Intraclass variance |
| $\tau_1$ | Within-class variance for variable $X_1$ |
| $\sigma$ | Random error |
| $N_{\text{Iter}}$ | Number of iterations in biokNN |
| $\mu_{\text{class}}$ | Average number of observations per class |
| $\sigma_{\text{class}}$ | Variance of the number of observations per class |
| $\alpha$ | Weighting parameter in biokNN |
| $k$ | Number of neighbors in biokNN |

### 3.4. Missing values pattern generation

There are three main assumptions regarding the mechanism that generates the missing values in a dataset, namely, missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR) (Schafer, 1997). MCAR assumes that the pattern of missing values does not depend on either the observed or the unobserved data. This scenario occurs because of errors in the data collection process (Razavi-Far et al., 2014). In an MAR scenario, the missing data depend on the observed values to some extent. Finally, the MNAR scenario arises when the pattern of missing values depends on the value of the variables in the observed dataset. In practice, missing value patterns are usually between MAR and MNAR (Wulff & Ejlskov, 2017).

In this study, we focus on an assumption of an MCAR generation pattern in both the target and explanatory variables, while the class variable is considered to be complete. However, our experiments show that the results are consistent with an MAR scenario. To generate MCAR patterns, we divide the generated dataset randomly into subsets, assuming that each value has the same probability to be missing.

## 4. Results

In this section, the results for the imputation performance of the proposed method are presented and analyzed. First, the imputation performance is compared with those of the benchmark methods by a simulation study. The performance is compared both in terms of the raw difference of the imputed values from the original generated data and in terms of the inference performance. The inference performance is compared by fitting a multilevel model and comparing the bias of the obtained estimates and the coefficient of determination. Second, three benchmark instances are used to test imputation accuracy. The analysis is conducted using R 3.6.3 and run on a 3 GHz Intel X5450 processor, with 24 GB RAM. An R package containing the biokNN procedure is available at https://CRAN.R-project.org/package=biokNN and it is described in Cubillos (2021).

### 4.1. Benchmark imputation methods

To evaluate the performance of the biokNN method, we compare its imputation accuracy with benchmark imputation methods that are summarized in Table 2. The mean imputation method (mean) imputes the missing value $x_{ip}$ by assigning it the mean value of the variable $p$. The $k$-nearest neighbors (knn) method assigns the imputed value the average of the neighbors based on the Euclidean distance of the observations. In the case of missing values of other variables, it uses the mean value of the variable.

The remaining imputation methods are based on chained equations processes in which the imputed values are obtained by estimations

**Table 2**
Description of the benchmark methods from the R package mice.

| | |
|---|---|
| mean | Unconditional mean imputation |
| knn | $k$-nearest neighbors |
| pmm | Predictive mean matching |
| 2l.norm | Level 1 normal heteroscedastic |
| 2l.lmer | Level 1 normal homoscedastic, lmer |
| 2l.pan | Level 1 normal homoscedastic, pan |
| 2lonly.mean | Level 2 class mean |
| 2l.jomo | Level 1 normal homoscedastic, jomo |

**Table 3**
Root mean square error (RMSE) of the benchmark methods and the proposed algorithm (multi.imp) on the simulated dataset, depending on the variance of the size of the classes.

| | $\sigma_{\text{size}} = 0$ | | $\sigma_{\text{size}} = 12$ | | $\sigma_{\text{size}} = 25$ | | $\sigma_{\text{size}} = 50$ | |
|---|---|---|---|---|---|---|---|---|
| Method | RMSE | diff % | RMSE | diff % | RMSE | diff % | RMSE | diff % |
| biokNN | **0.710** | — | **0.713** | — | **0.668** | — | **0.708** | — |
| knn | **0.710** | 0.01 | 0.757 | 5.8 | 0.795 | 16.0 | 0.821 | 13.8 |
| 2lonly.mean | 0.793 | 10.50 | 0.839 | 15.0 | 0.839 | 20.4 | 0.879 | 19.4 |
| 2l.norm | 0.844 | 15.93 | 0.853 | 16.4 | 0.794 | 15.9 | 0.777 | 8.9 |
| 2l.lmer | 0.848 | 16.30 | 0.886 | 19.5 | 0.769 | 13.1 | 0.767 | 7.6 |
| 2l.pan | 0.863 | 17.73 | 0.848 | 15.9 | 0.781 | 14.5 | 0.779 | 9.1 |
| pmm | 0.891 | 20.36 | 0.982 | 27.4 | 0.952 | 29.8 | 0.818 | 13.4 |
| 2l.jomo | 0.935 | 24.12 | 1.023 | 30.3 | 0.929 | 28.2 | 0.946 | 25.2 |
| mean | 1.018 | 30.32 | 1.011 | 29.4 | 0.985 | 32.2 | 0.990 | 28.4 |

**Table 4**
Root mean square error (RMSE) of the benchmark methods and the proposed algorithm (biokNN) on the simulated dataset. Results are shown for different levels of missing values $M$ and intraclass correlation $I$.

| $M$ | $I$ | biokNN | knn | 2l.norm | 2l.pan | 2l.lmer | 2lonly | 2l.jomo | mean | pmm |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.1 | 0.3 | 0.78 | **0.77** | 0.92 | 0.96 | 0.98 | 1.07 | 1.05 | 1.07 | 1.13 |
| | 0.5 | 0.77 | **0.76** | 0.90 | 0.94 | 0.95 | 1.05 | 1.02 | 1.07 | 0.98 |
| | 0.7 | 0.74 | **0.73** | 0.86 | 0.90 | 0.91 | 1.00 | 0.98 | 1.06 | 0.91 |
| | 0.9 | 0.71 | **0.69** | 0.85 | 0.75 | 0.86 | 0.92 | 0.82 | 1.06 | 0.96 |
| 0.3 | 0.3 | **0.77** | 0.78 | 1.03 | 1.04 | 0.97 | 0.90 | 1.32 | 0.93 | 1.20 |
| | 0.5 | **0.75** | 0.76 | 1.00 | 1.01 | 0.95 | 0.87 | 1.27 | 0.93 | 1.14 |
| | 0.7 | **0.71** | 0.73 | 0.96 | 0.97 | 0.90 | 0.82 | 1.17 | 0.93 | 1.10 |
| | 0.9 | **0.71** | 0.71 | 0.84 | 0.86 | 0.84 | 0.79 | 0.93 | 1.02 | 0.89 |
| 0.5 | 0.3 | **0.77** | 0.82 | 0.97 | 0.98 | 1.01 | 0.99 | 1.09 | 0.99 | 1.02 |
| | 0.5 | **0.75** | 0.80 | 0.97 | 0.93 | 0.99 | 0.95 | 1.06 | 1.00 | 1.07 |
| | 0.7 | **0.72** | 0.76 | 0.93 | 0.90 | 0.94 | 0.90 | 1.00 | 1.00 | 1.00 |
| | 0.9 | **0.66** | 0.69 | 0.82 | 0.80 | 0.85 | 0.79 | 0.89 | 1.02 | 0.86 |

where each variable takes its turn in being regressed on the other variables (Wulff & Ejlskov, 2017). These methods are implemented using the R package mice (Groothuis-Oudshoorn & Van Buuren, 2011). The predictive mean matching (pmm) method iteratively imputes missing values from selected known values in a given dimension using linear regressions.

The following benchmark methods incorporate the multilevel structure explicitly into the imputation method. In joint modeling approaches, a single multilevel model is specified for all variables with missing data, and it is implemented in R by two packages: pan (2l.pan) (Schafer & Zhao, 2016) and jomo (2l.jomo) (Quartagno & Carpenter, 2016). In the fully conditional specification model, missing data are imputed separately for each variable, and this has been implemented in mice using two functions that depend on the package used to specify the multilevel model: 2l.norm and 2l.lmer (Hox & Roberts, 2011). Finally, the model 2lonly.mean imputes the values by using the mean of the classes.

### 4.2. Comparison metrics

We base the comparison of the proposed method with the benchmark imputation methods on three main metrics. First, the imputation accuracy is compared by computing the root mean squared error (RMSE), which compares the imputed values directly with the values from the observed dataset:

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{M}|} \sum_{(i,p)\in\mathcal{M}} (w_{ip} - x_{ip})^2} \tag{11}$$

The true parameters of the generating model are known from the simulation. Thus, the bias of the estimated parameters after fitting a multilevel regression model can be directly compared. This comparison gives an estimate of how much the imputation method is affecting the inference process, which is done after the imputation step. To do that, we specify a varying-slope fit of a regression multilevel model using the package lme4 in R. This package uses maximum likelihood estimation to obtain estimates of the multilevel regression coefficients (Bates et al., 2014). If we take the true value of a model parameter to be $\theta$, then the percentage bias of the estimate $\hat{\theta}$ is given by

$$\text{Bias (\%)} = \frac{\theta - \hat{\theta}}{\theta} \times 100 \tag{12}$$

The final metric estimates the goodness-of-fit of the multilevel model, which can be interpreted as the variance explained by the entire model, including both fixed and random effects. The coefficient of determination for the multilevel setting, $R^2$, can be obtained from

$$R^2 = \frac{\tau_0^2 + \tau_1^2}{\tau_0^2 + \tau_1^2 + \sigma^2} \tag{13}$$

### 4.3. Simulation results

The simulation procedure is described as follows. First, we simulate datasets with a multilevel structure using random samples from a varying-slope model. Each dataset contains three variables: the target variable, an independent variable, and the class variable. Then, we randomly remove observations from both the target and the independent variables, assuming these to be MCAR with a fixed percentage of missing values. The imputation performance of the methods is measured by computing the RMSE between the observed and imputed datasets. All results shown in this subsection are average results over 100 simulations, and all variables are normalized to have unit standard deviation. The method parameters ($k$ and $\alpha$) are selected as the minima of the respective ranges $k \in \{10, 20, 30\}$ and $\alpha \in \{0.7, 0.8, 0.9\}$, based on preprocessing tests.

Table 3 shows the RMSE and the percentage difference of the benchmark methods with the proposed biokNN method, considering different variabilities between the size of the classes $\sigma_{\text{size}}$. We consider a mean size of $\mu_{\text{size}} = 25$, a missing value rate of $M = 0.3$, and an interclass correlation of $I = 0.9$. For all cases, the biokNN method shows the best imputation accuracy, with an average difference of 6.6% compared with the best benchmark method. The case in which all classes have the same number of observations ($\sigma_{\text{size}} = 0$) exhibits the lower difference with the best benchmark method (knn). The greatest difference with the best benchmark method is found at a level of $\sigma_{\text{size}} = 25$, with a difference of 13.1% compared with the 2l.lmer method.

To assess the effect of the level of missing values $M$ and the intraclass correlation $I$, Table 4 shows the RMSE results for different combinations of the two variables, assuming $\mu_{\text{size}} = 25$ and $\sigma_{\text{size}} = 0$. When the effect of $I$ is taken into account in the proposed method, for all three scenarios of missing values rates, the imputation accuracy increases as $I$ increases. From a low presence of multilevel structure ($I = 0.3$) to a high level ($I = 0.9$), biokNN improves its results by 10.2% on average. On the other hand, a comparison with the benchmark methods shows that the proposed method has better imputation accuracy for higher missing rates ($M = 0.3$ and $M = 0.5$), while the knn method gives better results in the case of low missing rates ($M = 0.1$)

In addition to comparing the error difference between the observed and imputed values based on the RMSE, the ability to provide adequate inference results is tested. Table 5 shows the $R^2$ and the percentage

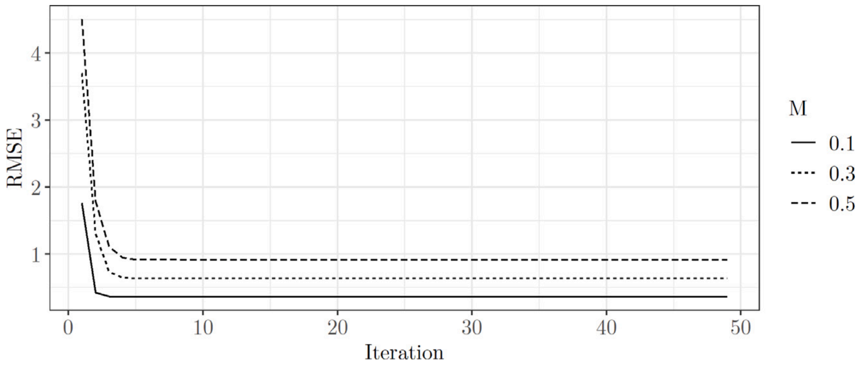

**Fig. 1.** RMSE over the iterations of the algorithm for a generated dataset. Results are shown for three missing rates $M = \{0.1, 0.3, 0.5\}$.

**Table 5**
Comparison of final fitted multilevel models including the percentage bias of the estimates and $R^2$.

| | | Percentage bias (%) | | | |
|---|---|---|---|---|---|
| Method | $R^2$ | $\hat{\beta}_0$ | $\hat{\tau}_0$ | $\hat{\beta}_1$ | $\hat{\tau}_1$ |
| biokNN | 0.915 | 3.79 | 3.55 | 0.54 | 3.04 |
| 2l.norm | 0.913 | 5.88 | 4.05 | 1.04 | 4.21 |
| 2l.pan | 0.901 | 5.84 | 6.08 | 1.32 | 6.41 |
| 2.lmer | 0.901 | 5.82 | 5.93 | 1.23 | 6.33 |
| knn | 0.898 | 5.86 | 5.41 | 1.02 | 7.16 |
| pmm | 0.882 | 6.92 | 4.02 | 1.06 | 22.85 |
| 2lonly.mean | 0.873 | 5.42 | 17.72 | 0.81 | 17.78 |
| 2l.jomo | 0.864 | 3.75 | 25.92 | 1.03 | 30.44 |
| mean | 0.748 | 4.89 | 16.86 | 15.10 | 18.75 |

bias of four of the regression parameters of the multilevel model using $M = 0.3$, $I = 0.9$, $\mu_{\text{size}} = 25$, and $\sigma_{\text{size}} = 0$. On average, biokNN provides a better goodness-of-fit and reduced bias of the estimates. In terms of goodness-of-fit, in contrast to the results obtained by considering the RMSE, the results are much closer compared with the best benchmark model (2l.norm). However, biokNN is able to obtain estimates that are on average 30.9% less biased that those given by the 2l.norm method. In this case, the methods that do not take account of the class variables in the imputation (knn, pmm, 2lonly.mean, and mean) show a reduced overall goodness-of-fit, and their estimates show a much higher bias compared with biokNN, especially in the estimation of the slope variance between classes ($\hat{\tau}_1$).

In terms of inference performance after imputation, our method gave better results in terms of goodness-of-fit and parameter estimation compared with the FCS and JM approaches. In our simulation, FCS and JM gave very similar results, in spite of their different theoretical basis. The advantage of our method over FCS and JM is its simplicity and transparency. Our method does not require any distributional assumptions or model specification. Also, its flexibility allows the incorporation of prior knowledge into the imputation by selection of the weighting parameter. This allows users to decide how much importance the imputation is giving to the observations within the same class in a transparent way, avoiding the black-box problem of other imputation methods.

Overall, the simulation results suggest that biokNN can provide gains in imputation accuracy for most of the multilevel configurations tested, particularly in cases with higher missing value rates and unbalanced classes. In terms of multilevel inference, biokNN gives competitive results compared with the best FCS imputation methods (2l.norm, 2l.pan) and is able to obtain unbiased estimates for both main and random effects. By contrast, the methods that ignore the multilevel structure of the data show inadequate fits and higher bias in the estimates, particularly in the case of variability of the slope among classes.

### 4.4. Results on benchmark datasets

We select five representative benchmark datasets to test the effectiveness of the proposed methodology: two from the UCI repository (machine and sleepstudy), one from the datasets available in the R package lme4 (cbpp), and two from Snijders and Bosker (2011) (ml-book red and soep). The selection was made in order to have different configurations regarding intraclass correlation, number of variables, number of classes, and variability between the sizes of the classes. For each dataset, observations were made missing values randomly under the assumption of MCAR by taking $M$ from 0.1 to 0.7, and then the RMSE was computed based on the imputed values.

Table 6 shows the results obtained by the best four models. For each dataset, we show the number of observations ($N$), the number of variables ($p$), number of classes ($Q$), intraclass correlation ($I$), average observations per class ($\mu_{\widehat{\text{size}}}$), and standard deviation of the number of observations per class ($\sigma_{\widehat{\text{size}}}$). For each dataset, we selected one categorical variable to be the class variable and one variable to be the continuous variable with missing values. We remove the rest of categorical variables if they are present since our method is designed for continuous variables only.

Consistently with the results obtained from the simulation, biokNN has the best imputation accuracy for the dataset with the highest variability in the size of the classes. This result is shown by the machine dataset, for which, excluding the case of low missing rate ($M = 0.1$), biokNN is able to give the lowest RMSE values. The lowest performance is shown for the dataset where all classes have the same number of observations (sleepstudy), for which both the knn and 2l.pan methods gave most accurate results.

### 4.5. Convergence

The proposed method is based on a iterative first-order method and we propose the selection of a number $N$ of iterations as a stop criteria. The convergence of the method depends on the characteristics of the dataset with missing values and the quality of the start solution of the method, which is set to be a random imputation. In our experiments, we observe that the convergence of our algorithm is relatively fast and it is often met after a few iterations. For simulated data, we find that the selection of $N = 10$ iterations is sufficient, even when increasing the amount of missing values in the same dataset. As an example, in Fig. 1 we show the RMSE values over 50 iterations of the biokNN algorithm for a simulated dataset using $I = 0.9$, $Q = \mu_{class} = 25$. We present the results for three missing values rates $M = \{0.1, 0.3, 0.5\}$. Convergence in the three cases is met relatively fast, being faster in the case with lower missing rates. Our results are in concordance with results shown by Bertsimas et al. (2017) in the single-objective version of the problem regarding the speed of the convergence of the first-order method for imputation.

**Table 6**
RMSE of the four best benchmark methods and the proposed algorithm on five benchmark datasets.

| Dataset | $N$ | $p$ | $Q$ | $I$ | $\hat{\mu_{size}}$ | $\hat{\sigma_{size}}$ | $M$ | biokNN | 2l.norm | 2l.pan | knn |
|---|---|---|---|---|---|---|---|---|---|---|---|
| machine | 209 | 9 | 30 | 0.37 | 6.9 | 6.9 | 0.1 | **0.52** | 0.62 | 0.60 | 0.51 |
| | | | | | | | 0.3 | **0.65** | 0.83 | 0.66 | 0.69 |
| | | | | | | | 0.5 | **0.67** | 0.76 | 0.73 | 0.74 |
| | | | | | | | 0.7 | **0.63** | 0.73 | 0.71 | 0.68 |
| sleepstudy | 180 | 2 | 18 | 0.11 | 10 | 0 | 0.1 | 0.97 | 0.97 | 0.93 | **0.78** |
| | | | | | | | 0.3 | 0.96 | 0.96 | 0.93 | **0.83** |
| | | | | | | | 0.5 | 0.97 | 0.97 | 0.99 | **0.84** |
| | | | | | | | 0.7 | 1.02 | 1.09 | 1.12 | **0.91** |
| cbpp | 56 | 3 | 15 | 0.01 | 3.73 | 0.69 | 0.1 | **0.92** | 1.72 | 1.11 | **0.92** |
| | | | | | | | 0.3 | **0.99** | 1.41 | 1.30 | 1.04 |
| | | | | | | | 0.5 | **1.00** | 1.91 | 1.42 | 1.06 |
| | | | | | | | 0.7 | **0.97** | 1.85 | 1.47 | 1.03 |
| mlbook red | 3758 | 2 | 259 | 0.11 | 17.8 | 7.15 | 0.1 | **0.30** | 0.40 | 0.37 | 0.37 |
| | | | | | | | 0.3 | **0.51** | 0.67 | 0.61 | 0.61 |
| | | | | | | | 0.5 | **0.63** | 0.84 | 0.78 | 0.78 |
| | | | | | | | 0.7 | **0.77** | 1.08 | 0.96 | 0.97 |
| soep | 6024 | 2 | 23 | 0.02 | 261.9 | 72.9 | 0.1 | **0.29** | 0.40 | 0.39 | 0.34 |
| | | | | | | | 0.3 | **0.61** | 0.77 | 0.71 | 0.71 |
| | | | | | | | 0.5 | **0.72** | 0.97 | 0.92 | 0.86 |
| | | | | | | | 0.7 | **0.85** | 1.14 | 1.05 | 0.99 |

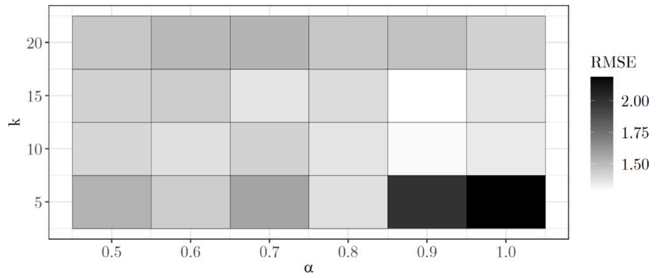

**Fig. 2.** Illustration of the calibration step. In this example the best selection of parameters is $k = 15$ and $\alpha = 0.9$.

### 4.6. Parameter calibration

The performance of the biokNN algorithm depends on the selection of two main parameters: the relative weight given to the two objectives ($\alpha$) and the number of neighbors in the kNN part of the function ($k$). The best selection of parameters depends on the dataset and may be affected by the size, number of classes, intraclass correlation, and amount of missing values. The more simple way to determine this parameters is using a grid search and selecting the lowest error metric. This can be done by extracting a percentage of extra missing values on the dataset and compare the errors produced by the different configurations of parameters. In Fig. 2 we illustrate the parameter selection phase for a generated dataset ($M = 0.1, I = 0.9$) by extracting a 10% of extra missing values. The best configuration of parameters in this example is $\alpha = 0.9$ and $k = 15$.

In Fig. 3 we explore the sensitivity of the selection of parameters in the RMSE for 100 simulated datasets with different parameter configurations. In the right side of the figure we show the effect of varying the parameter $k$ for $\alpha = \{0.5, 0.7, 0.9\}$ and in the left side we show the effect of varying the parameter $\alpha$ for $k = \{5, 10, 15\}$. In general, we observe large variability in the RMSE values. However, the lowest RMSE values are concentrated for lower values of the number of neighbor ($k < 20$) and higher values of the weight parameter ($\alpha > 0.7$).

### 4.7. Computational runtime

In this section, we compare the computational run time of our method to other five benchmark imputation methods. In Fig. 4 we show the average run time over 100 simulated datasets using 25 observations per class, and increasing the number of classes from 10 to 100. With this, we compare the computational run times for datasets with 250 to 2500 observations. The most simple methods that do not include the multilevel structure into the imputation (mean, pmm, 2lonly.mean) are almost instantaneous since they require little computation. The method 2l.pan also show fast computational times. The method 2l.norm, shows a linear increase in the computational time that increases from 3 to 21 s on average in our experiments. Finally, the biokNN method shows concave increase in time which overpass the time of the 2l.norm method with datasets with more than 70 classes.

Finally, in Fig. 5 we show the effect of the amount of missing data on computational runtime when increasing the number of classes. Results are the average runtime in seconds over 100 simulated datasets with 25 observations per class for three missing rates $M = \{0.1, 0.3, 0.5\}$. We observe that computational runtimes are higher for higher missing values in all cases. The relative difference in runtimes between the three missing rates increases as we increase the number of classes. For instance, the difference between $M = 0.1$ and $M = 0.5$ is about 2 s for datasets with 20 classes, while the difference is almost 14 s when considering 70 classes.

## 5. Discussion and future research

In the biokNN method it is not required to specify an imputation model. This is a major advantage when researchers need to run several different multilevel models after imputation. Current model-based imputation methods need researchers to specify an imputation model that is at least as general as the analysis models (Grund et al., 2018). When the research questions demand different multilevel analysis models, the number of such models grows rapidly. Models may have different varying intercepts and slopes, and one model might need to account for mediation (Zhang et al., 2009), while others might include cross-level interactions (Aguinis & Culpepper, 2015), model the variance (Lester et al., 2021), or account for endogeneity (Antonakis et al., 2021). Combining such model structures into one imputation model can be a daunting task even for the most seasoned researchers. The biokNN method circumvents this problem completely by not requiring any model specification in the imputation process. This makes biokNN very attractive to researchers who need several or even just a few complex multilevel analysis models to answer their research questions.

There are some limitations to the proposed methodology that are worth noting. The first of these regards parameter tuning. Selection of the parameters of the method can be challenging in the case of imputation of datasets with large numbers of missing values, since

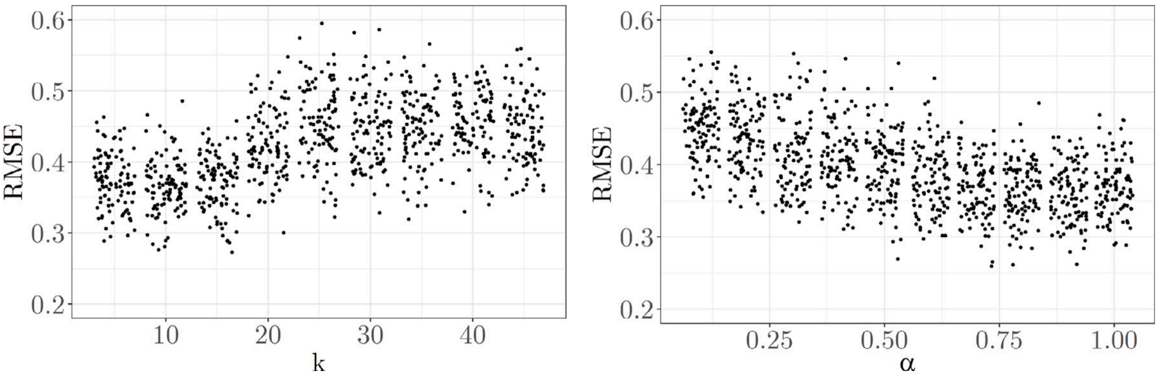

**Fig. 3.** Sensitivity of the parameter selection on the RMSE for 100 simulated datasets.

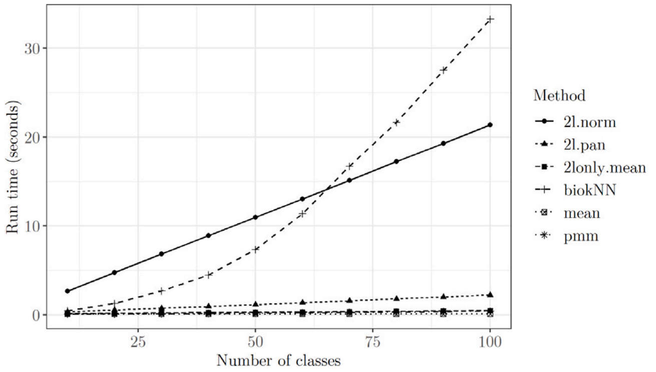

**Fig. 4.** Comparison of the computational runtimes of five benchmark imputation methods.

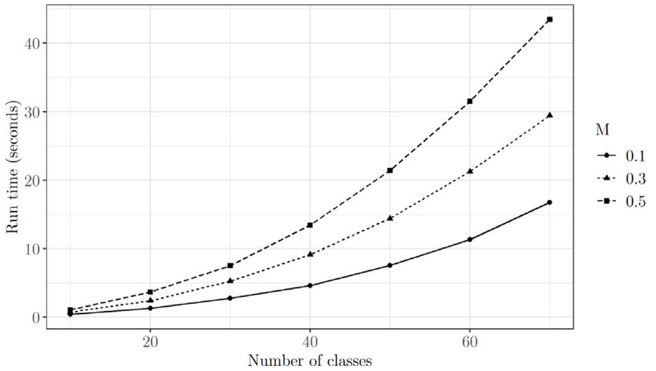

**Fig. 5.** Effect of the amount of missing values in the computational runtimes of the biokNN method.

the selection has to be made based on the remaining complete observations. For large datasets, a calibration preprocessing step may be computationally expensive compared with other methods. Future work could design a calibration method that can rapidly provide reliable parameters for the model. Extensions to handle missing data in the class variable and to handle categorical variables should also be addressed.

The biokNN method can be extended from a bi-objective to a multi-objective method to take other features of the structure of the data into consideration. For instance, more weighted objective terms can be added for datasets with two or more levels of hierarchy. Future work could also include the use of different optimization methods to provide imputation values instead of the kNN method. For instance, Bertsimas et al. (2017) have explored the use of kNN, support vector machines, and decision-tree-based optimization methods in a similar fashion for single-level imputations.

## 6. Conclusion

We proposed an imputation method to handle missing values in the presence of data with multilevel structures. The problem was described as an optimization problem in which we aimed to minimize two objectives: the dissimilarity between the $k$-nearest neighbors and the observations within the same clusters. We proposed an algorithm to solve the imputation problem. To test the imputation accuracy of the proposed method, we compared its results with those of the most common imputation methods used for multilevel imputation. The methods were compared both by simulation and by using benchmark datasets. The results showed that the proposed method gives better imputation accuracy and can reduce the bias of multilevel models, especially in the case of high missing rates and high intraclass correlation.

## CRediT authorship contribution statement

**Maximiliano Cubillos:** Conceptualization, Methodology, Data curation, Writing – original draft. **Sanne Wøhlk:** Supervision, Reviewing. **Jesper N. Wulff:** Writing – review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This project was supported by the AUFF NOVA grant for the project ‘Opening the black box: Making Machine Learning interpretable for Organizational Research’ [grant number AUFF-E-2019-9-3]

## References

Aguinis, H., & Culpepper, S. A. (2015). An expanded decision-making procedure for examining cross-level interaction effects with multilevel modeling. *Organizational Research Methods*, *18*(2), 155–176. http://dx.doi.org/10.1177/1094428114563618.

Al-Helali, B., Chen, Q., Xue, B., & Zhang, M. (2021). A new imputation method based on genetic programming and weighted KNN for symbolic regression with incomplete data. *Soft Computing*, *25*, 5993–6012. http://dx.doi.org/10.1007/s00500-021-05590-y.

Andridge, R. R. (2011). Quantifying the impact of fixed effects modeling of clusters in multiple imputation for cluster randomized trials. *Biometrical Journal*, *53*(1), 57–74. http://dx.doi.org/10.1002/bimj.201000140.

Antonakis, J., Bastardoz, N., & Rönkkö, M. (2021). On ignoring the random effects assumption in multilevel models: Review, critique, and recommendations. *Organizational Research Methods*, *24*(2), 443–483. http://dx.doi.org/10.1177/1094428119877457.

Arias-Castro, E., & Donoho, D. L. (2009). Does median filtering truly preserve edges better than linear filtering? *The Annals of Statistics*, *37*(3), 1172–1206. http://dx.doi.org/10.1214/08-AOS604.

Awawdeh, S., Faris, H., & Hiary, H. (2022). EvoImputer: An evolutionary approach for missing data imputation and feature selection in the context of supervised learning. *Knowledge-Based Systems*, *236*, Article 107734. http://dx.doi.org/10.1016/j.knosys.2021.107734.

Barner, K., & Arce, G. R. (2003). *Nonlinear signal and image processing: Theory, methods, and applications*. CRC Press, https://www.routledge.com/Nonlinear-Signal-and-Image-Processing-Theory-Methods-and-Applications/Barner-Arce/p/book/9780849314278.

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2014). Fitting linear mixed-effects models using lme4. arXiv preprint arXiv:1406.5823.

Bertsekas, D. (1999). *Nonlinear programming*. Belmont, Massachusets, USA: Athena Scientific.

Bertsimas, D., Pawlowski, C., & Zhuo, Y. D. (2017). From predictive methods to missing data imputation: An optimization approach. *Journal of Machine Learning Research*, *18*(1), 7133–7171, https://jmlr.org/papers/v18/17-073.html.

Black, A. C., Harel, O., & Betsy McCoach, D. (2011). Missing data techniques for multilevel data: Implications of model misspecification. *Journal of Applied Statistics*, *38*(9), 1845–1865. http://dx.doi.org/10.1080/02664763.2010.529882.

Cai, Z., Heydari, M., & Lin, G. (2006). Iterated local least squares microarray missing value imputation. *Journal of Bioinformatics and Computational Biology*, *4*(05), 935–957. http://dx.doi.org/10.1142/s0219720006002302.

Carpenter, J., & Kenward, M. (2012). *Multiple imputation and its application*. Chichester: Wiley.

Caruana, R. (2001). A non-parametric EM-style algorithm for imputing missing values. In *AISTATS*.

Caselles, V., Sapiro, G., & Chung, D. (2000). Vector median filters, inf-sup operations, and coupled PDE's: Theoretical connections. *Journal of Mathematical Imaging and Vision*, *12*, 109–119. http://dx.doi.org/10.1023/A:1008310305351.

Cubillos, M. (2021). *biokNN: A bi-objective imputation method for multilevel data in R*. Aarhus University, https://pure.au.dk/portal/files/214627979/biokNN.pdf.

Drechsler, J. (2015). Multiple imputation of multilevel missing data—Rigor versus simplicity. *Journal of Educational and Behavioral Statistics*, *40*(1), 69–95. http://dx.doi.org/10.3102/1076998614563393.

Enders, C. K., Mistler, S. A., & Keller, B. T. (2016). Multilevel multiple imputation: A review and evaluation of joint modeling and chained equations imputation. *Psychological Methods*, *21*(2), 222–240. http://dx.doi.org/10.1037/met0000063.

Garciarena, U., & Santana, R. (2017). An extensive analysis of the interaction between missing data types, imputation methods, and supervised classifiers. *Expert Systems with Applications*, *89*, 52–65. http://dx.doi.org/10.1016/j.eswa.2017.07.026.

George, G., Oommen, R. M., Shelly, S., Philipose, S. S., & Varghese, A. M. (2018). A survey on various median filtering techniques for removal of impulse noise from digital image. In *2018 conference on emerging devices and smart systems* (pp. 235–238). http://dx.doi.org/10.1109/ICEDSS.2018.8544273.

Goldstein, H., Carpenter, J. R., & Browne, W. J. (2014). Fitting multilevel multivariate models with missing data in responses and covariates that may include interactions and non-linear terms. *Journal of the Royal Statistical Society. Series A. Statistics in Society*, *177*(2), 553–564. http://dx.doi.org/10.1111/rssa.12022.

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

Grund, S., Lüdtke, O., & Robitzsch, A. (2016). Multiple imputation of missing covariate values in multilevel models with random slopes: A cautionary note. *Behavior Research Methods*, *48*(2), 640–649. http://dx.doi.org/10.3758/s13428-015-0590-3.

Grund, S., Lüdtke, O., & Robitzsch, A. (2018). Multiple imputation of missing data for multilevel models: Simulations and recommendations. *Organizational Research Methods*, *21*(1), 111–149. http://dx.doi.org/10.1177/1094428117703686.

Horton, N. J., & Kleinman, K. P. (2007). Much Ado About Nothing: A comparison of missing data methods and software to fit incomplete data regression models. *The American Statistician*, *61*(1), 79–90. http://dx.doi.org/10.1198/000313007X172556.

Hox, J., & Roberts, J. K. (Eds.), (2011). *Handbook of advanced multilevel analysis*. New York: Psychology Press.

Jiang, C., & Yang, Z. (2015). In D. -S. Huang, & K. Han (Eds.), *CKNNI: An improved KNN-based missing value handling technique* (pp. 441–452). Cham: Springer.

Kim, H., Golub, G. H., & Park, H. (2005). Missing value estimation for DNA microarray gene expression data: Local least squares imputation. *Bioinformatics*, *21*(2), 187–198. http://dx.doi.org/10.1093/bioinformatics/bth499.

Kim, K. Y., Kim, B. J., & Yi, G. S. (2004). Reuse of imputed data in microarray analysis increases imputation efficiency. *BMC Bioinformatics*, *5*(1), 1–9. http://dx.doi.org/10.1186/1471-2105-5-160.

Lan, Q., Xu, X., Ma, H., & Li, G. (2020). Multivariable data imputation for the analysis of incomplete credit data. *Expert Systems with Applications*, *141*, Article 112926. http://dx.doi.org/10.1016/j.eswa.2019.112926.

Lester, H. F., Cullen-Lester, K. L., & Walters, R. W. (2021). From nuisance to novel research questions: Using multilevel models to predict heterogeneous variances. *Organizational Research Methods*, *24*(2), 342–388. http://dx.doi.org/10.1177/1094428119887434.

Lin, W. -C., & Tsai, C. -F. (2020). Missing value imputation: A review and analysis of the literature (2006–2017). *Artificial Intelligence Review*, *53*(2), 1487–1509. http://dx.doi.org/10.1007/s10462-019-09709-4.

Lin, W. C., Tsai, C. F., & Zhong, J. R. (2022). Deep learning for missing value imputation of continuous data and the effect of data discretization. *Knowledge-Based Systems*, Article 108079. http://dx.doi.org/10.1016/j.knosys.2021.108079.

Little, R. J., & Rubin, D. B. (2019). *Statistical analysis with missing data*. Hoboken, NJ: Wiley.

Lüdtke, O., Robitzsch, A., & Grund, S. (2017). Multiple imputation of missing data in multilevel designs: A comparison of different strategies. *Psychological Methods*, *22*(1), 141–165. http://dx.doi.org/10.1037/met0000096.

Mistler, S. A. (2015). *Multilevel multiple imputation: An examination of competing methods* (Doctoral dissertation), Arizona State University.

Pan, R., Yang, T., Cao, J., Lu, K., & Zhang, Z. (2015). Missing data imputation by K nearest neighbours based on grey relational structure and mutual information. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *43*(3), 614–632. http://dx.doi.org/10.1007/s10489-015-0666-x.

Purwar, A., & Singh, S. K. (2015). Hybrid prediction model with missing value imputation for medical data. *Expert Systems with Applications*, *42*(13), 5621–5631. http://dx.doi.org/10.1016/j.eswa.2015.02.050.

Quartagno, M., & Carpenter, J. R. (2016). jomo: A package for multilevel joint modelling multiple imputation. R package.

Rachdi, M., Laksaci, A., Kaid, Z., Benchiha, A., & Al-Awadhi, F. A. (2021). k-Nearest neighbors local linear regression for functional and missing data at random. *Statistica Neerlandica*, *75*(1), 42–65. http://dx.doi.org/10.1111/stan.12224.

Raghunathan, T. E., Lepkowski, J. M., Van Hoewyk, J., & Solenberger, P. (2001). A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey Methodology*, *27*(1), 85–96.

Razavi-Far, R., Zio, E., & Palade, V. (2014). Efficient residuals pre-processing for diagnosing multi-class faults in a doubly fed induction generator, under missing data scenarios. *Expert Systems with Applications*, *41*(14), 6386–6399. http://dx.doi.org/10.1016/j.eswa.2014.03.056.

Schafer, J. L. (1997). *Analysis of incomplete multivariate data*. Boca Raton, FL: CRC Press.

Schafer, J., & Zhao, J. (2016). pan: Multiple imputation for multivariate panel or clustered data. R package.

Sefidian, A. M., & Daneshpour, N. (2019). Missing value imputation using a novel grey based fuzzy c-means, mutual information based feature selection, and regression model. *Expert Systems with Applications*, *115*, 68–94. http://dx.doi.org/10.1016/j.eswa.2018.07.057.

Snijders, T. A., & Bosker, R. J. (2011). *Multilevel analysis: An introduction to basic and advanced multilevel modeling*. London: SAGE.

Song, S., & Sun, Y. (2020). Imputing various incomplete attributes via distance likelihood maximization. In *KDD 20: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 535–545). http://dx.doi.org/10.1145/3394486.3403096.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., & Altman, R. B. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics*, *17*(6), 520–525. http://dx.doi.org/10.1093/bioinformatics/17.6.520.

Tukey, J. W. (1977). *Exploratory data analysis* (pp. 581–582). Reading, Ma, 688: Addision-Wesley.

Tutz, G., & Ramzan, S. (2015). Improved methods for the imputation of missing data by nearest neighbor methods. *Computational Statistics & Data Analysis*, *90*, 84–99. http://dx.doi.org/10.1016/j.csda.2015.04.009.

Wright, S. J. (2015). Coordinate descent algorithms. *Mathematical Programming*, *151*(1), 3–34. http://dx.doi.org/10.1007/s10107-015-0892-3.

Wulff, J. N., & Ejlskov, L. (2017). Multiple imputation by chained equations in praxis: Guidelines and review. *Electronic Journal of Business Research Methods*, *15*(1), 41–56, http://www.ejbrm.com/volume15/issue1.

Zadeh, L. (1963). Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, *8*(1), 59–60. http://dx.doi.org/10.1109/TAC.1963.1105511.

Zhang, X., Song, X., Wang, H., & Zhang, H. (2008). Sequential local least squares imputation estimating missing value of microarray data. *Computers in Biology and Medicine*, *38*(10), 1112–1120. http://dx.doi.org/10.1016/j.compbiomed.2008.08.006.

Zhang, Z., Zyphur, M. J., & Preacher, K. J. (2009). Testing multilevel mediation using hierarchical linear models: Problems and solutions. *Organizational Research Methods*, *12*(4), 695–719. http://dx.doi.org/10.1177/1094428108327450.