Journal of Data Science 22 (3), 456–468
July 2024

DOI: 10.6339/24-JDS1135
Statistical Data Science

Predictive Mean Matching Imputation Procedure Based on Machine Learning Models for Complex Survey Data

Sixia Chen[1,*] and Chao Xu[1]

[1]*University of Oklahoma Health Sciences Center, 801 NE 13th ST, Oklahoma City, OK, 73104, United States*

Abstract

Missing data is a common occurrence in various fields, spanning social science, education, economics, and biomedical research. Disregarding missing data in statistical analyses can introduce bias to study outcomes. To mitigate this issue, imputation methods have proven effective in reducing nonresponse bias and generating complete datasets for subsequent analysis of secondary data. The efficacy of imputation methods hinges on the assumptions of the underlying imputation model. While machine learning techniques such as regression trees, random forest, XGBoost, and deep learning have demonstrated robustness against model misspecification, their optimal performance may necessitate fine-tuning under specific conditions. Moreover, imputed values generated by these methods can sometimes deviate unnaturally, falling outside the normal range. To address these challenges, we propose a novel Predictive Mean Matching imputation (PMM) procedure that leverages popular machine learning-based methods. PMM strikes a balance between robustness and the generation of appropriate imputed values. In this paper, we present our innovative PMM approach and conduct a comparative performance analysis through Monte Carlo simulation studies, assessing its effectiveness against other established methods.

Keywords *imputation; missing data; nonresponse bias*

1 Introduction

Missing data happens frequently in practice including biomedical study, educational study, economics, and sample surveys. According to Akinbami et al. (2022), the response rates of interviewed sample and examined sample for 2017–2020 National Health Nutrition and Examination Survey (NHANES) are only 51% and 46.9%. According to 2022 Summary Data Quality Report, the median rates for all states and territories of 2022 Behavioral Risk Factor Surveillance System (BRFSS) is only 45%. Simply ignoring missing data in statistical analysis may lead to biased results (Little and Rubin, 2019). Furthermore, causal inference (Imbens and Rubin, 2015), latent variable model (Loehlin, 2004), measurement error model (Fuller, 2009), and data integration problem (Yang and Kim, 2020b; Chen et al., 2022) can be regarded as special cases of missing data problems (Kim and Shao, 2021). In practice, missing data can be generally classified into two broad categories: unit non-response and item non-response. Unit non-response happens when some subjects are absent for the entire survey or study. Item non-response happens when some subjects only miss part of the survey or study. Unit non-response is usually handled by using inverse weighting procedure and item non-response is usually handled by imputation pro-

*Corresponding author. Email: sixia-chen@ouhsc.edu or chao-xu@ouhsc.edu.

Received November 22, 2023; Accepted April 16, 2024

cedures, see Kim and Shao (2021) for a detailed discussion of different methods. We will focus on item non-response in this paper.

The idea for imputation procedures is to fill the missing values in the data file by using some predictive values from some model with the respondents. Commonly used imputation methods include regression imputation (Zhang, 2016), hot-deck imputation approaches (Rao and Shao, 1992; Andridge and Little, 2010), nearest neighbor imputation approach (Chen and Shao, 2000; Yang and Kim, 2019), predictive mean matching (PMM) imputation (Little, 1988; Heitjan and Little, 1991; Yang and Kim, 2020a), fractional imputation approaches (Kim and Fuller, 2004; Kim, 2011), and multiple imputation (Rubin, 1996, 2018). The validity of each imputation method depends on the underlying model assumptions. To further improve the robustness against the underlying model assumptions, nonparametric imputation methods including kernel smoothing method (Cheng, 1994), spline method (Chen et al., 2022), and many others were developed. However, the nonparametric imputation methods suffer from the curse of dimensionality. Recently, machine learning based imputation methods including regression tree (Burgette and Reiter, 2010; Rahman and Islam, 2011), random forest (Shah et al., 2014; Tang and Ishwaran, 2017), XGboost (Deng and Lumley, 2023; Qiao et al., 2018), support vector machines (Mallinson and Gammerman, 2003; Aydilek and Arslan, 2013), and deep neural networks (Lin et al., 2020; Chen and Xu, 2023) have been developed for handling the complex nonlinear structure and high dimensionality in missing data analysis. Even though the machine learning methods have been shown to protect against the failure of underlying model assumptions, they all depend on certain model structures including tuning parameters and they are not robust against the outliers. Optimal and proper selection of tuning parameters for statistical estimation of population parameters such as population mean and quantiles has not been developed in existing literature.

Compared with other imputation methods, PMM generates imputed values which are selected from existing true values of respondents, so the imputed values are always within the desired range. In addition, PMM is more robust against the model misspecification and outliers compared with other parametric imputation methods, and it does not suffer from the curse of dimensionality as the nonparametric imputation methods. However, the existing PMM method is based on the parametric model and the validity of it depends on the Lipschitz continuity condition (Yang and Kim, 2020a). To improve the robustness, Chen et al. (2021) proposed multiply robust PMM method with complex survey data by using multiple imputation models simultaneously. However, if none of the models was correctly specified and the Lipschitz continuity condition was violated, their proposed estimators would be biased. The PMM method based on machine learning methods has not been developed in existing literature. We hypothesize that the impact of tuning for machine learning methods might be small with using PMM method, and PMM method based on machine learning methods is more robust compared with existing PMM method based on parametric model methods. In this paper, we fill the important research gap by developing a novel predictive mean matching (PMM) imputation procedure based on machine learning models including K nearest neighbor (KNN), generalized additive model (GAM), support vector machine (SVM), XGboost, and deep neural network methods. In addition, we evaluate the performance of different methods by Monte Carlo simulation studies. Our proposed methods are developed in general settings with any complex sampling designs. User friendly computational code has also been developed for other researchers to use.

Our paper was organized as follows. Section 2 introduces basic setups and notations. Proposed method is discussed in Section 3. Section 4 contains Monte Carlo simulation study. Computational aspects including the selection of tuning parameters, computational speed, and example

code are contained in Section 5. Discussion of our findings and future research are presented in Section 6.

2 Basic Setups

Consider a finite population with independent and identically distributed copies

$$\mathcal{F}_N = \{(\mathbf{x}_i, y_i), i = 1, 2, \ldots, N\},$$

where $\mathbf{x}$ is the covariate vector with dimension p and y is the study variable of interest. Assume they have been generated from the following super-population outcome regression model:

$$y_i = m(\mathbf{x}_i) + \epsilon_i, \quad i = 1, 2, \ldots, N, \tag{1}$$

where $m(\mathbf{x}_i)$ is assumed to be unknown and the ϵ_i's are assumed to be mutually independent random variables such that $E(\epsilon_i|\mathbf{x}_i) = 0$ and $V(\epsilon_i|\mathbf{x}_i) = \sigma^2$. For simplicity, we assumed the variance structure is homoscedastic in above model, but our proposed method can be naturally extended to the heteroscedastic scenario. Given the finite population $\mathcal{F}_N$, suppose a random sample s is selected from some probability sampling design with sample size n. The corresponding design weight is assumed to be w_i for $i = 1, 2, \ldots, N$. We assume the covariate vector $\mathbf{x}_i$ is fully observed for $i \in s$ and the study variable of interest y_i is subject to missingness with response indicator denoted as δ_i such that $\delta_i = 1$ if unit i is observed and $\delta_i = 0$ otherwise. The missing mechanism is assumed to be missing at random (MAR) (Little and Rubin, 2019):

$$\Pr(\delta_i = 1|\mathbf{x}_i, y_i) = \Pr(\delta_i = 1|\mathbf{x}_i). \tag{2}$$

The missing mechanism is assumed to follow the positivity assumption such that $\Pr(\delta_i = 1|\mathbf{x}_i) > c$ for a positive constant c with probability 1.

We denote s_r as the set of sampled subjects with $\delta_i = 1$, s_m as the set of sampled subjects with $\delta_i = 0$, so $s = s_r \cup s_m$. Suppose we are interested in estimating the population mean of y, which is $\theta_0 = E(y)$. The existing PMM procedure can be described as follows. One first assumes a parametric regression working model $m(\mathbf{x}_i) = m(\mathbf{x}_i; \boldsymbol{\beta})$, with $\boldsymbol{\beta}$ as the vector of unknown parameters. Then one can fit the above regression model by using $i \in s_r$ and survey weight to obtain the score $\hat{m}_i = m(\mathbf{x}_i; \hat{\boldsymbol{\beta}})$ for all subjects $i \in s$, where $\hat{\boldsymbol{\beta}}$ is the solution of the weighted estimating equation:

$$\sum_{i \in s_r} w_i \{y_i - m(\mathbf{x}_i; \hat{\boldsymbol{\beta}})\} \frac{\partial m(\mathbf{x}_i; \hat{\boldsymbol{\beta}})}{\partial \boldsymbol{\beta}} = 0. \tag{3}$$

After that, the imputed value y_i^* for subject $i \in s_m$ can be obtained as $y_i^* = y_j$ with j as the index of the nearest-neighbor of unit i from s_r in terms of minimizing the distance defined by the score $\hat{m}_i$. The PMM method is valid only if the following Lipschitz continuity condition (Yang and Kim, 2020a) is satisfied: $d\{m(\mathbf{x}_i), m(\mathbf{x}_j)\} \leqslant Ad(\mathbf{x}_i, \mathbf{x}_j)$ for some distance function d and positive constant A. However, this condition may not be satisfied for some commonly used models such as quadratic models. Finally, the PMM estimator of θ_0 can be written as:

$$\hat{\theta}_{PMM} = \frac{1}{\hat{N}} \left(\sum_{i \in s_r} w_i y_i + \sum_{i \in s_m} w_i y_i^* \right), \tag{4}$$

where $\hat{N} = \sum_{i \in s} w_i$. Asymptotic properties including consistency and asymptotic normality have been considered in Yang and Kim (2020a).

3 Proposed Method

Instead of using parametric model for constructing PMM estimator, we propose using the modern machine learning methods to improve the robustness against the model misspecification. Commonly used machine learning methods include Generalized Additive Model, K-Nearest Neighbors (KNN) Algorithm, Regression Tree, Random Forest, XGboost, Support Vector Machine (SVM), and Deep Neural Networks. Furthermore, to improve the robustness, one can also use super learner (Van der Laan et al., 2007; Polley and Van der Laan, 2010) to combine multiple machine learning models. Suppose we consider the working machine learning model $E(y|\mathbf{x}) = \tilde{m}(\mathbf{x})$. Then one can fit the above machine learning model by using the set of respondents s_r with design weight w_i. Generalized Additive Model is a nonparametric additive learning model and its fitting can be done by backfitting (Hastie, 2017). The KNN algorithm was presented by Peterson (2009). The optimal partitioning for classification and regression trees have been discussed by Chou (1991) and Steinberg and Colla (2009). Random forest is an extension of regression tree and the corresponding algorithm has been discussed in Breiman (2001). The optimal algorithm for XGboost was introduced by Chen and Guestrin (2016). SVM model can be estimated by using the algorithms presented by Noble (2006) and Hearst et al. (1998). Deep neural network models can be fitted by common optimizers include Stochastic Gradient Descent (Bottou, 2010; Das et al., 2016), Adam (Kingma and Ba, 2014), and RMSprop (Hinton et al., 2012). The selection of tuning parameters will be discussed in Section 4.

Even though the machine learning methods are somewhat robust against the incorrect underlying parametric model assumptions, they all need certain regularity conditions for the underlying models to produce consistent model fitting, see Toth and Eltinge (2011), Wager and Athey (2018), Farrell et al. (2021), among others. PMM used weaker regularity conditions and the fitted values $\hat{y}_i = \hat{\tilde{m}}(\mathbf{x}_i)$ for the whole sample s can be obtained from the previous fitted model. Then, the imputed value for subject $i \in s_m$ can be written as $y_i^* = y_j$, with j as the index of the nearest-neighbor of subject $i \in s_r$ such that $d(\hat{y}_j, \hat{y}_i) \leqslant d(\hat{y}_{j'}, \hat{y}_i)$ for any $j' \in s_r$, where d is some distance function. We consider Euclidean distance function in this paper. Finally, the machine learning based PMM estimator can be written as:

$$\hat{\theta}_{ML} = \frac{1}{\hat{N}}\left(\sum_{i\in s_r} w_i y_i + \sum_{i\in s_m} w_i y_i^*\right). \tag{5}$$

The asymptotic properties such as consistency and asymptotic normality for some machine learning based methods can be established by using similar results in Toth and Eltinge (2011), Wager and Athey (2018), Farrell et al. (2021), and Yang and Kim (2020a). For variance estimation, one can use the replication variance estimation method discussed by Yang and Kim (2020a).

4 Simulation Study

We generated $B = 200$ Monte Carlo samples of finite populations with population size $N = 20{,}000$ from the following three outcome regression models:

(M1). Linear model structure: $y_i = 1 + x_{1,i} + x_{2,i} + x_{3,i} + x_{4,i} + \epsilon_i$ for $i = 1, 2, \ldots, N$, where $x_{1,i}$, $x_{2,i}$, $x_{3,i}$, $x_{4,i}$, and ϵ_i are generated from the standard normal distribution and they are all independent.

(M2). Non-linear model structure: $y_i = 1+x_{1,i}^2+x_{3,i}x_{4,i}+x_{1,i}x_{2,i}+x_{3,i}^3+x_{4,i}^4+\epsilon_i$ for $i = 1, 2, \ldots, N$, where $x_{1,i}$, $x_{2,i}$, $x_{3,i}$, $x_{4,i}$, and ϵ_i are generated from the standard normal distribution and they are all independent.

(M3). Hierarchical non-linear model structure: First layer modeling is the following:

$$a_{11,i} = \log\{1 + \exp(\alpha_{111} + \alpha_{112}x_{1,i} + \alpha_{113}x_{2,i} + \alpha_{114}x_{3,i} + \alpha_{115}x_{4,i})\} + \epsilon_{11,i}, \quad (6)$$

$$a_{21,i} = \log\{1 + \exp(\alpha_{211} + \alpha_{212}x_{1,i} + \alpha_{213}x_{2,i} + \alpha_{214}x_{3,i} + \alpha_{215}x_{4,i})\} + \epsilon_{21,i}, \quad (7)$$

and

$$a_{31,i} = \log\{1 + \exp(\alpha_{311} + \alpha_{312}x_{1,i} + \alpha_{313}x_{2,i} + \alpha_{314}x_{3,i} + \alpha_{315}x_{4,i})\} + \epsilon_{31,i}, \quad (8)$$

where α_{111}, α_{112}, α_{113}, α_{114}, α_{115}, α_{211}, α_{212}, α_{213}, α_{214}, α_{215}, α_{311}, α_{312}, α_{313}, α_{314}, and α_{315} are generated from a uniform distribution with range from -2 to 2. $\epsilon_{11,i}$, $\epsilon_{21,i}$, and $\epsilon_{31,i}$ are generated from the standard normal distribution. Second layer modeling is the following:

$$a_{12,i} = \log\{1 + \exp(\alpha_{121} + \alpha_{122}a_{11,i} + \alpha_{123}a_{21,i} + \alpha_{124}a_{31,i})\} + \epsilon_{12,i}, \quad (9)$$

$$a_{22,i} = \log\{1 + \exp(\alpha_{221} + \alpha_{222}a_{11,i} + \alpha_{223}a_{21,i} + \alpha_{224}a_{31,i})\} + \epsilon_{22,i}, \quad (10)$$

and

$$a_{32,i} = \log\{1 + \exp(\alpha_{321} + \alpha_{322}a_{11,i} + \alpha_{323}a_{21,i} + \alpha_{324}a_{31,i})\} + \epsilon_{32,i}, \quad (11)$$

where α_{121}, α_{122}, α_{123}, α_{124}, α_{221}, α_{222}, α_{223}, α_{224}, α_{321}, α_{322}, α_{323}, and α_{324} are generated from a uniform distribution with range from -2 to 2. $\epsilon_{12,i}$, $\epsilon_{22,i}$, and $\epsilon_{32,i}$ are generated from the standard normal distribution. Final layer modeling is the following:

$$y_i = \beta_0 + \beta_1 a_{12,i} + \beta_2 a_{22,i} + \beta_3 a_{32,i} + \epsilon_i, \quad (12)$$

for $i = 1, 2, \ldots, N$, where β_0, β_1, β_2, and β_3 are generated from a uniform distribution with range from -2 to 2 and ϵ_i is generated from the standard normal distribution.

The response indicator δ_i is generated from a Bernoulli distribution with the following probability:

$$\Pr(\delta_i = 1|\mathbf{x}_i) = 0.1 + 0.9\frac{\exp(\alpha_0 + \alpha_1 x_{1,i} + \alpha_2 x_{2,i} + \alpha_3 x_{3,i} + \alpha_4 x_{4,i})}{1 + \exp(\alpha_0 + \alpha_1 x_{1,i} + \alpha_2 x_{2,i} + \alpha_3 x_{3,i} + \alpha_4 x_{4,i})}, \quad (13)$$

where $\mathbf{x}_i = (x_{1,i}, x_{2,i}, x_{3,i}, x_{4,i})$ and $(\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4) = (-0.36, 1, 1, 1, 1)$ which leads to about 50% response rate. Given each finite population, a simple random sample without replacement is selected with sample size $n = 500$. Our parameter of interest is the population mean of y. We consider the following approaches: (1). Naive estimator (NAIVE) by using the sample mean of the respondents; (2). Regression estimator (REG) by assuming linear regression model; (3). Predictive mean matching (PMM) estimator by using linear regression model; (4). PMM method based on generalized additive model (GAM); (5). PMM method based on K nearest neighbor method (KNN); (6). PMM method based on support vector machine (SVM); (7). PMM method based on XGboost method (XGB); (8). PMM methods based on deep neural network methods (DNN-aL-bN) with a number of layers (3, 4, and 5) and b number of nodes (50, 100, and 200). The following Rectified Linear Unit (ReLU) activation function was considered in the simulation study: $f(x) = \max(0, x)$. This function is renowned for its superior computational stability, maintaining model flexibility without compromise. As a result, it is widely recognized as the standard choice in contemporary neural network literature, as detailed in Goodfellow et al. (2016). Tuning parameters of KNN, SVM, and XGB were selected by using 10-fold cross validation method. The list of tuning parameters is presented in Table 4.

We calculate the following Monte Carlo relative bias (RB), relative standard error (RSE), and relative root mean squared error for all above methods:

$$RB = \frac{B^{-1}\sum_{k=1}^{B}\hat{\theta}_k - \theta_0}{|\theta_0|}, \tag{14}$$

$$RSE = \frac{\{(B-1)^{-1}\sum_{k=1}^{B}(\hat{\theta}_k - \bar{\hat{\theta}})^2\}^{1/2}}{|\theta_0|}, \tag{15}$$

and

$$RRMSE = \left(RB^2 + RSE^2\right)^{1/2}, \tag{16}$$

where $\hat{\theta}_k$ is the estimator based on the k-th Monte Carlo sample, θ_0 is our parameter of interest (e.g., population mean of y), and $\bar{\hat{\theta}} = B^{-1}\sum_{k=1}^{B}\hat{\theta}_k$. Note that $\hat{\theta}_k$ denotes any one of the above estimators considered in the simulation study.

The results are presented from Table 1 to Table 3 for three models (M1), (M2), and (M3). Under linear model (M1), NAIVE estimator had the largest RB and RRMSE since it suffered from large nonresponse bias. REG estimator had the smallest RB, RSE, and RRMSE besides the RB of DNN-3L-200N since it used the correct model assumption without the loss of information. PMM and GAM estimators had comparable small RB, RSE, and RRMSE since they both used correct model assumptions. KNN method had large RB and RRMSE due to the inaccurate modeling process and the loss of information. XGB method had a larger RB than SVM and most of the DNN methods, and it had a smaller RSE. The performance of DNN methods depends on the number of layers and the nodes. The RB ranges from −0.34% to 9.17%. DNN-5L-200N, DNN-4L-100N, DNN-3L-100N, and DNN-3L-200N had the best results. Under nonlinear model (M2), NAIVE and REG estimators had the largest RB and RRMSE since they

Table 1: Relative bias (RB) (%), Relative standard error (RSE) (%), and Relative root mean squared error (RRMSE) (%) of different methods under model (M1).

Method	RB	RSE	RRMSE
NAIVE	107.21	12.57	107.95
REG	−0.51	11.69	11.70
PMM	1.07	13.12	13.16
GAM	0.61	13.15	13.16
KNN	15.73	14.67	21.51
SVM	4.82	13.03	13.89
XGB	9.04	12.82	15.69
DNN-5L-50N	9.17	15.29	17.83
DNN-5L-100N	6.68	15.04	16.45
DNN-5L-200N	−0.76	16.77	16.78
DNN-4L-50N	2.06	15.72	15.85
DNN-4L-100N	0.75	15.90	15.92
DNN-4L-200N	−1.92	16.35	16.46
DNN-3L-50N	7.00	15.22	16.75
DNN-3L-100N	−1.30	16.60	16.65
DNN-3L-200N	−0.34	15.76	15.76

Table 2: Relative bias (RB) (%), Relative standard error (RSE) (%), and Relative root mean squared error (RRMSE) (%) of different methods under model (M2).

Method	RB	RSE	RRMSE
NAIVE	18.65	13.83	23.22
REG	−20.24	17.59	26.81
PMM	−1.28	18.54	18.58
GAM	−5.55	9.14	10.69
KNN	−10.09	9.20	13.66
SVM	−3.55	10.34	10.93
XGB	−3.95	11.09	11.78
DNN-5L-50N	0.55	11.92	11.93
DNN-5L-100N	−2.27	11.24	11.47
DNN-5L-200N	−1.22	11.45	11.51
DNN-4L-50N	−5.55	9.93	11.37
DNN-4L-100N	−0.69	11.23	11.25
DNN-4L-200N	2.02	11.58	11.76
DNN-3L-50N	−0.21	11.47	11.47
DNN-3L-100N	−0.89	12.07	12.10
DNN-3L-200N	−0.04	11.65	11.65

Table 3: Relative bias (RB) (%), Relative standard error (RSE) (%), and Relative root mean squared error (RRMSE) (%) of different methods under model (M3).

Method	RB	RSE	RRMSE
NAIVE	38.16	9.09	39.22
REG	3.80	9.52	10.24
PMM	4.14	10.47	11.26
GAM	4.34	10.83	11.67
KNN	7.80	10.62	13.17
SVM	2.86	10.99	11.36
XGB	1.76	9.96	10.11
DNN-5L-50N	2.95	8.95	9.43
DNN-5L-100N	1.50	9.33	9.45
DNN-5L-200N	2.30	9.39	9.67
DNN-4L-50N	0.34	8.75	8.75
DNN-4L-100N	2.59	9.36	9.71
DNN-4L-200N	2.32	9.33	9.61
DNN-3L-50N	0.47	9.10	9.11
DNN-3L-100N	1.17	9.25	9.33
DNN-3L-200N	1.43	9.19	9.31

suffered from a large selection bias with using the incorrect model assumptions. PMM method still had a small RB due to the robustness of PMM, but it had large RSE and RRMSE compared with other machine learning based methods. GAM method had large RB and RRMSE due to the incorrect specification of underlying model. KNN had large bias and RRMSE due to the inaccurate modeling process and the loss of information. SVM, XGB, DNN-4L-50N had similar performance, while other DNN methods had better performance. Other DNN methods showed stable performance by producing the RB ranging from −2.27% to −0.04% and the RRMSE ranging from 11.25% to 12.10%. DNN-3L-200N had the best performance in terms of RB. DNN-4L-100N had be the best performance in terms of RRMSE. Under hierarchical non-linear model (M3), NAIVE method had the largest RB and RRMSE. KNN method had the second largest RB and RRMSE. REG, PMM, and GAM had comparable results which were worse than other machine learning methods. SVM, XGB, and many DNN methods had comparable results. DNN-4L-50N had the best results with RB 0.34% and RRMSE 8.75%.

5 Computational Aspect

The machine learning methods KNN, SVM, and XGB were tuned using R packages parsnip, workflows, recipes, and dials. The key hyper-parameters we tuned for these methods are listed in Table 4. For each of these methods, 30 combination sets of tuning hyper-parameters were searched, based on 10-fold cross validation, to select an optimum set by minimizing the predictive root mean squared error (RMSE). The default search range provided by the R packages for the hyper-parameters were used. Given the tuning hyper-parameters, the final model was derived. For the deep learning model, since we did not find a package for hyper-parameter tuning, we tried 5 learning rate (0.1, 0.01, 0.001, 0.0001, and 0.00001), in combination with 3 choices of number of layers (3, 4, 5) and number of nodes (50, 100, 200). The best result from the 5 learning rate was reported. Python modules including keras, sklearn, numpy, and pandas were used for deep learning.

Table 4: List of tuning parameters involved in machine learning approaches.

Approach	Parameter	Definition
KNN	neighbors	number of neighbors to consider
	weight_func	type of kernel function used to weight distances between samples
	dist_power	parameter used in calculating Minkowski distance
SVM	rbf_sigma	a positive number for radial basis function
	cost	cost of predicting a sample within or on the wrong side of the margin
	margin	the epsilon in the SVM insensitive loss function
XGB	tree_depth	maximum depth of the tree
	trees	number of trees contained in the ensemble
	learn_rate	learning rate
	mtry	number of selected predictors
	min_n	minimum number of data points in a node

```
//Let fn = input filename with full path
>dat <- read_rds(fn)
>head(dat)
          [,1] [,2]        [,3]        [,4]        [,5]        [,6] [,7]
[1,] 10.5681487    0  0.66334506  0.53520659  0.43884418 -1.51644942   40
[2,] -9.1545868    0 -1.01711358 -0.54766613  0.89750604 -0.44738307   40
[3,] -6.1845442    0 -1.02766226 -0.03151297  0.02747651 -0.68613398   40
[4,] -0.3040334    1 -0.06401128  0.09116087 -0.54777198 -1.34625481   40
[5,] -5.2300663    1 -1.86555488  0.43973496 -0.55454071 -0.48680561   40
[6,] -1.3552546    1  0.22969701  0.25733036  0.80921284 -0.03276383   40

//f_ML is the function to call our algorithim using different method specified by modeling_method
//The choice of modeling_method includes "Naive", "Reg", "PMM1", "GAM", "XGBOOST", "KNN", "SVM".
> res=f_ML(dat,"SVM")

//The function returns two items: the estimate of variable of interest (VOI) and the individual value
//of VOI after imputation
> str(res)
List of 2
 $ : num 4.16
 $ : tibble [500 × 1] (S3: tbl_df/tbl/data.frame)
  ..$ .pred: num [1:500] 9.469 -0.162 -0.94 4.689 -3.342 ...
```

Figure 1: An example running code.

We have uploaded all of our R and Python scripts to Github at https://github.com/xu1912/PMM-imputation. Given an example data of 500 samples, from which 232 are missing response y, we can get an estimate of true y by calling our wrapped function F_ML with an imputation method. Please see the example R code in Figure 1. The Python example is available on the Github site. We recorded the computation time for 200 simulations using different methods. The Naive, regression, PMM, and GAM had the 4 shortest running times, which were <1 min for 200 simulations in all 3 models. SVM, KNN, and XGBoost needed around 250 mins, 260 mins, and 760 mins respectively for 200 simulations in all 3 models. The running time of the Deep Learning method was not stable in all models. For model 1, it took about 551 mins to complete 45 scenarios (3 choices of layers $\times$ 3 choices of nodes $\times$ 5 learning rates). For model 3, it took only about 328 mins to run 45 scenarios. In practice, we may only need one or a few runs, which means about 2–3 mins to try 45 combinations of hyper-parameters for deep learning method.

6 Discussion

In this paper, we have introduced innovative predictive mean matching (PMM) imputation procedures grounded in modern machine learning methods. Our proposed techniques are highly versatile, capable of application to data files featuring diverse and complex survey designs. Our proposed methods can also be applied to causal inference and data integration research areas since they are special scenarios of missing data problems. We conducted a limited Monte Carlo study to compare the performance of different imputation methods.

Our findings revealed that PMM methods based on advanced machine learning approaches, including Support Vector Machines (SVM), Extreme Gradient Boosting (XGB), and Deep Neural Networks (DNN), outperformed other methods in scenarios characterized by complex nonlinear model structures. The parametric regression imputation approach exhibited good performance when underlying assumptions of the parametric model were met.

The optimal selection of tuning parameters for the DNN method, encompassing factors like the activation function, number of layers, and number of nodes, remains an open question. The traditional 10-fold cross-validation method may not yield the optimal estimator concerning the minimization of mean squared error, indicating the need for further research in this direction.

Moreover, we observed instability in the computational time associated with DNN methods, warranting additional research to enhance stability. An intriguing avenue for exploration involves comparing our proposed methods with other machine learning approaches, such as super-learner.

Our proposed methods offer considerable practical appeal, particularly given the inherent challenges of discerning the underlying model mechanism in real-world data. Additionally, by utilizing existing donor values as imputed values, our approach effectively sidesteps the issue of generating unrealistic imputations. Moreover, our methods demonstrate robustness against outliers and are not susceptible to the pitfalls associated with high-dimensional data—a phenomenon commonly referred to as the "curse of dimensionality."

Lastly, our future plans include evaluating the performance of our proposed methods through real data applications, which we intend to document in a separate manuscript. This endeavor aims to provide practical insights into the applicability and effectiveness of our methods in real-world scenarios. Additionally, we intend to assess the effectiveness of our proposed methods in high-dimensional settings through a comprehensive evaluation involving both Monte Carlo simulation studies and real-world data applications.

Supplementary Material

Predictive mean matching imputation procedure based on machine learning models, using R and Python: https://github.com/xu1912/PMM-imputation/tree/main

Funding

Dr. Sixia Chen was partially supported by the Oklahoma Shared Clinical and Translational Resources (U54GM104938) with an Institutional Development Award (IDeA) from NIGMS. The content is solely the responsibility of the authors and does not necessarily represent official views of the National Institutes of Health or the Indian Health Service. Part of the computing for this project was performed at the OU Supercomputing Center for Education & Research (OSCER) at the University of Oklahoma (OU).

References

Akinbami LJ, Chen TC, Davy O, Ogden CL, Fink S, Clark J, et al. (2022). National health and nutrition examination survey, 2017–March 2020 prepandemic file: Sample design, estimation, and analytic guidelines.

Andridge RR, Little RJ (2010). A review of hot deck imputation for survey non-response. *International Statistical Review*, 78(1): 40–64. https://doi.org/10.1111/j.1751-5823.2010.00103.x

Aydilek IB, Arslan A (2013). A hybrid method for imputation of missing values using optimized fuzzy c-means with support vector regression and a genetic algorithm. *Information Sciences*, 233: 25–35. https://doi.org/10.1016/j.ins.2013.01.021

Bottou L (2010). Large-scale machine learning with stochastic gradient descent. In: Lechevallier Y, Saporta G (eds.), *Proceedings of COMPSTAT'2010: 19th International Conference on*

Computational Statistics, Paris, France, August 22–27, 2010 Keynote, Invited and Contributed Papers, 177–186. Springer.

Breiman L (2001). Random forests. *Machine Learning*, 45: 5–32. https://doi.org/10.1023/A:1010933404324

Burgette LF, Reiter JP (2010). Multiple imputation for missing data via sequential regression trees. *American Journal of Epidemiology*, 172(9): 1070–1076. https://doi.org/10.1093/aje/kwq260

Chen J, Shao J (2000). Nearest neighbor imputation for survey data. *Journal of Official Statistics*, 16(2): 113.

Chen S, Haziza D, Stubblefield A (2021). A note on multiply robust predictive mean matching imputation with complex survey data. *Survey Methodology*, 47(1): 215–223.

Chen S, Xu C (2023). Handling high-dimensional data with missing values by modern machine learning techniques. *Journal of Applied Statistics*, 50(3): 786–804. https://doi.org/10.1080/02664763.2022.2068514

Chen S, Yang S, Kim JK (2022). Nonparametric mass imputation for data integration. *Journal of Survey Statistics and Methodology*, 10(1): 1–24. https://doi.org/10.1093/jssam/smaa036

Chen T, Guestrin C (2016). Xgboost: A scalable tree boosting system. In: Krishnapuram B et al. (eds.), *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

Cheng PE (1994). Nonparametric estimation of mean functionals with data missing at random. *Journal of the American Statistical Association*, 89(425): 81–87. https://doi.org/10.1080/01621459.1994.10476448

Chou PA (1991). Optimal partitioning for classification and regression trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(04): 340–354. https://doi.org/10.1109/34.88569

Das D, Avancha S, Mudigere D, Vaidynathan K, Sridharan S, Kalamkar D, et al. (2016). Distributed deep learning using synchronous stochastic gradient descent. arXiv preprint: https://arxiv.org/abs/1602.06709.

Deng Y, Lumley T (2023). Multiple imputation through XGBoost. *Journal of Computational and Graphical Statistics*, 33(2): 352–363. https://doi.org/10.1080/10618600.2023.2252501

Farrell MH, Liang T, Misra S (2021). Deep neural networks for estimation and inference. *Econometrica*, 89(1): 181–213. https://doi.org/10.3982/ECTA16901

Fuller WA (2009). *Measurement Error Models*. John Wiley & Sons.

Goodfellow I, Bengio Y, Courville A (2016). *Deep Learning*. MIT Press.

Hastie TJ (2017). Generalized additive models. In: Chambers JM, Hastie TJ (eds.), *Statistical Models in S*, 249–307. Routledge.

Hearst MA, Dumais ST, Osuna E, Platt J, Scholkopf B (1998). Support vector machines. *IEEE Intelligent Systems & Their Applications*, 13(4): 18–28. https://doi.org/10.1109/5254.708428

Heitjan DF, Little RJ (1991). Multiple imputation for the fatal accident reporting system. *Journal of the Royal Statistical Society. Series C. Applied Statistics*, 40(1): 13–29.

Hinton G, Srivastava N, Swersky K (2012). Neural networks for machine learning. Lecture 6a. Overview of mini-batch gradient descent. *Cited on*, 14(8): 2.

Imbens GW, Rubin DB (2015). *Causal Inference in Statistics, Social, and Biomedical Sciences*. Cambridge University Press.

Kim JK (2011). Parametric fractional imputation for missing data analysis. *Biometrika*, 98(1): 119–132. https://doi.org/10.1093/biomet/asq073

Kim JK, Fuller W (2004). Fractional hot deck imputation. *Biometrika*, 91(3): 559–578. https://doi.org/10.1093/biomet/91.3.559

Kim JK, Shao J (2021). *Statistical Methods for Handling Incomplete Data*. CRC Press.

Kingma DP, Ba J (2014). Adam: A method for stochastic optimization. arXiv preprint: https://arxiv.org/abs/1412.6980.

Lin J, Li N, Alam MA, Ma Y (2020). Data-driven missing data imputation in cluster monitoring system based on deep neural network. *Applied Intelligence*, 50: 860–877. https://doi.org/10.1007/s10489-019-01560-y

Little RJ (1988). Missing-data adjustments in large surveys. *Journal of Business & Economic Statistics*, 6(3): 287–296. https://doi.org/10.1080/07350015.1988.10509663

Little RJ, Rubin DB (2019). *Statistical Analysis with Missing Data*, volume 793. John Wiley & Sons.

Loehlin JC (2004). *Latent Variable Models: An Introduction to Factor, Path, and Structural Equation Analysis*. Psychology Press.

Mallinson H, Gammerman A (2003). Imputation using support vector machines. *University of London Egham, UK: Department of Computer Science Royal Holloway*.

Noble WS (2006). What is a support vector machine? *Nature Biotechnology*, 24(12): 1565–1567. https://doi.org/10.1038/nbt1206-1565

Peterson LE (2009). K-nearest neighbor. *Scholarpedia*, 4(2): 1883. https://doi.org/10.4249/scholarpedia.1883

Polley EC, Van der Laan MJ (2010). Super learner in prediction.

Qiao L, Ran R, Wu H, Zhou Q, Liu S, Liu Y (2018). Imputation method of missing values for dissolved gas analysis data based on iterative KNN and XGBoost. In: *Proceedings of the 2018 International Conference on Algorithms, Computing and Artificial Intelligence*, 1–7.

Rahman MG, Islam MZ (2011). A decision tree-based missing value imputation technique for data pre-processing. In: Vamplew P, Stranieri A, Ong K-L, Christen P, Kennedy PJ (eds.), *The 9th Australasian Data Mining Conference: AusDM 2011*, 41–50. Australian Computer Society Inc.

Rao JN, Shao J (1992). Jackknife variance estimation with survey data under hot deck imputation. *Biometrika*, 79(4): 811–822. https://doi.org/10.1093/biomet/79.4.811

Rubin DB (1996). Multiple imputation after 18+ years. *Journal of the American Statistical Association*, 91(434): 473–489. https://doi.org/10.1080/01621459.1996.10476908

Rubin DB (2018). Multiple imputation. In: van Buuren S (ed.), *Flexible Imputation of Missing Data*, Second Edition, 29–62. Chapman and Hall/CRC.

Shah AD, Bartlett JW, Carpenter J, Nicholas O, Hemingway H (2014). Comparison of random forest and parametric imputation models for imputing missing data using mice: A caliber study. *American Journal of Epidemiology*, 179(6): 764–774. https://doi.org/10.1093/aje/kwt312

Steinberg D, Colla P (2009). Cart: classification and regression trees. In: Wu X, Kumar V (eds.), *The Top Ten Algorithms in Data Mining*, volume 9, 179.

Tang F, Ishwaran H (2017). Random forest missing data algorithms. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 10(6): 363–377.

Toth D, Eltinge JL (2011). Building consistent regression trees from complex sample data. *Journal of the American Statistical Association*, 106(496): 1626–1636. https://doi.org/10.1198/jasa.2011.tm10383

Van der Laan MJ, Polley EC, Hubbard AE (2007). Super learner. *Statistical Applications in*

Genetics and Molecular Biology, 6(1): 25.

Wager S, Athey S (2018). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523): 1228–1242. https://doi.org/10.1080/01621459.2017.1319839

Yang S, Kim JK (2019). Nearest neighbor imputation for general parameter estimation in survey sampling. In: Huynh KP, Jacho-Chávez DT, Tripathi G (eds.), *The Econometrics of Complex Survey Data*, volume 39, 209–234. Emerald Publishing Limited.

Yang S, Kim JK (2020a). Asymptotic theory and inference of predictive mean matching imputation using a superpopulation model framework. *Scandinavian Journal of Statistics*, 47(3): 839–861. https://doi.org/10.1111/sjos.12429

Yang S, Kim JK (2020b). Statistical data integration in survey sampling: A review. *Japanese Journal of Statistics and Data Science*, 3: 625–650. https://doi.org/10.1007/s42081-020-00093-w

Zhang Z (2016). Missing data imputation: Focusing on single imputation. *Annals of Translational Medicine*, 4(1): 9. https://doi.org/10.21037/atm-20-3623