Statistica Sinica **30** (2020), 1697-1722
doi:https://doi.org/10.5705/ss.202019.0122

REJOINDER

MISSING DATA, IMPUTATION AND REGRESSION TREES

Wei-Yin Loh[1], Qiong Zhang[2], Wenwen Zhang[3] and Peigen Zhou[1]

[1] *University of Wisconsin, Madison,* [2] *Clemson University and* [3] *Takeda*

Abstract: Missing data are a major hindrance to statistical analysis because standard methods require the missing values to be imputed first. AMELIA and MICE are two popular imputation methods, but their effectiveness has not been scrutinized in complex data. Loh et al. (2019) showed that these imputation methods are impractical in an application where the number of variables and the quantity of missing values are large. They proposed a GUIDE piecewise-constant regression tree as an alternative as it does not require imputation and can handle large numbers of variables. Little (2020) questioned the generality of their conclusions as well as the assumptions behind machine learning methods. This article responds to the criticisms and uses a large simulation experiment to compare the parameter estimation bias of GUIDE and MICE and the prediction accuracy of several model-based and machine learning regression algorithms after GUIDE and MICE imputation.

Key words and phrases: Machine learning, missing at random, prediction accuracy, regression forest.

1. Introduction

Parametric likelihood-based imputation has been the dominant approach to dealing with missing data for many years. The last decade, however, saw exciting developments from machine learning, such as matrix completion methods (Candès and Recht (2009)), that are motivated by problems in image analysis (Tomasi and Kanade (1993)), recommender systems (Koren, Bell and Volinsky (2009)), genomics (Chi et al. (2013); Cai, Cai and Zhang (2016)), and remote sensing (Jiang, Zhang and Qiao (2018)). Loh et al. (2019) (hitherto abbreviated as LECL) introduced another approach based on the GUIDE (Loh (2002, 2009)) regression tree algorithm and showed how it can be used to estimate a population mean for data from the Bureau of Labor Statistics Consumer Expenditure (CE) Survey. They demonstrated that GUIDE can fit regression models to a dependent (Y) variable *without* imputation of missing values in the predictor (X) variables.

They also showed that a GUIDE classification tree can estimate propensity scores for use in an inverse probability weighted (IPW) estimate, again *without* imputation of missing values in the X variables. Unlike likelihood-based imputation methods, such as AMELIA (Honaker, King and Blackwell (2011)) and MICE (van Buuren and Groothuis-Oudshoorn (2011)), GUIDE has no computational difficulties with large sample size, large number of variables, large amounts of missing data, and different types of missing value patterns.

Little (2020) doubted the extent to which general conclusions can be drawn from the study in LECL. He also questioned the assumptions behind machine learning models and their missing data mechanisms. The goals of this article are (i) to correct some misconceptions about tree methods, (ii) to clarify why and when imputation is needed and why it is unnecessary in GUIDE, (iii) to compare the estimation bias of linear models when MICE and GUIDE are used for imputation, and (iv) to evaluate the prediction accuracy of parametric and machine learning methods applied to missing data imputed by different techniques under different missingness mechanisms.

The remainder of this article is organized as follows. Section 2 corrects some common misconceptions of classification and regression tree methods. Section 3 recalls the original reasons for missing-value imputation and explains why it is not needed in GUIDE. Section 4 uses five simple examples to highlight the problem of estimation bias in linear models due to model misspecification and violation of the assumptions required for imputation by MICE. Section 5 uses 24 real data sets to study the effect of different imputation methods on the prediction accuracy of 9 parametric and machine learning methods under different missing-value mechanisms. Section 6 concludes the article with some remarks.

2. Misconceptions

A tree model has the same objective as a linear model, which is to estimate a regression function. Little (2020) claimed that tree models assume the true regression function to be a step function, stating, “categorization of continuous predictors assumes that the relationship with the outcome is a step function.” A tree model is a piecewise linear approximation to the true function, no less than a linear model is a linear approximation. There is no reason why a linear model cannot be used if the true model is not linear. Unlike a linear model, however, a piecewise-constant or piecewise-linear tree model is adaptive in that it typically converges pointwise to the true function as the sample size increases (Chaudhuri et al. (1994)). Little (2020) claimed that spline models are preferable for their continuity. Although continuity may be desirable in certain situations, spline models face great challenges when the number of predictor variables is large. Besides, continuity is relevant only for continuous variables. Section 5 below shows that the prediction accuracy of spline models, as represented by MARS (Friedman (1991)), may be poor.

Little (2020) stated that, "since LECL do not advance theoretical arguments in favor of their tree methods, the main basis for comparison of methods is their simulation study." Conditions for the asymptotic behavior of tree methods have been known for some time. Breiman et al. (1984) established Bayes risk consistency and Chaudhuri et al. (1994), Chaudhuri et al. (1995) and Chaudhuri and Loh (2002) gave conditions for uniform consistency over compact sets of the conditional mean and quantile function estimates. As for MICE, its author (van Buuren (2012, p. 249)) noted: "There is no clear theoretical rationale for convergence of the multivariate algorithm. The main justification of the MICE algorithm rests on simulation studies."

Little (2020) also claimed that "tree methods assume a missing not at random (MNAR) mechanism, because they include indicators of missingness of predictors as covariates." Indicators of missingness can be used as covariates by any method—they are not exclusive to tree models. Traditional imputation models tend not to use the indicators because they create bias in regression coefficients (Vach and Blettner (1991); Knol et al. (2010)). The reason GUIDE piecewise-constant tree and forest models do not require missingness assumptions on the covariates is simply that they do not require the missing values to be imputed.

3. Imputation

Parametric regression techniques require missing values in ordinal predictor variables to be imputed because the methods are inapplicable otherwise. Traditional imputation methods, however, themselves rely on parametric models to do their work. This is a chicken-and-egg situation, where a parametric model cannot be fitted without missing value imputation and an imputation method cannot produce imputations without fitting parametric models. Methods such as MICE get around the problem by initially imputing missing values with means and modes and then iteratively fitting parametric models to the data to update the imputed values one variable at a time.

GUIDE piecewise-constant regression tree and forest models do not require imputation of missing values in the predictor variables because their values are only used to define the splits in a tree. Let NA denote the missing value code. A split is defined by a condition "$X \in S$," where X is a predictor variable and S is a subset of values (possibly including NA) of X. Observations go to the left child node if and only if the condition is satisfied. If X is categorical, NA is given its own categorical level called "missing". Consequently, there are never missing values in categorical variables in GUIDE. If X is ordinal, the condition takes the form "$X \leq c$" or "$\{X \leq c\} \cup \{X = \texttt{NA}\}$", with c chosen to minimize the total sums of squared residuals in the left and right child nodes. The split "$\{X \leq c\}$" sends missing X values to the right whereas "$\{X \leq c\} \cup \{X = \texttt{NA}\}$" sends missing values to the left. The split "$\{X \leq -\infty\} \cup \{X = \texttt{NA}\}$" is interpreted as "$X = \texttt{NA}$". Thus missing X values are not imputed.

The CE data analyzed in LECL consisted of observations on about 600 variables from 4,609 individual respondents, called "consumer units" (CU). Treating INTRDVX, the amount of interest and dividend income, as the Y variable, LECL compared different estimation methods for the population mean of INTRDVX. About a third of the INTRDVX values in the data values were recorded as missing. In addition, 20 percent of the other variables had missing values, including 67 variables with more than 95 percent values missing.

Models that employ ordinal X variables are usually fitted in one of two ways. Either omit the ordinal variables with missing values or impute them first. In the CE data, omitting ordinal variables with missing values discards 20 percent of the variables. Although 80 percent of 600 is still a large number, some of the omitted variables are important for prediction. LECL found that imputing the missing values in 600 predictor variables cannot be performed with the MICE and AMELIA software, due to multicollinearity and other computational difficulties. Little (2020) claimed that "they couldn't get it to work" and mentioned stepwise, ridge, or LASSO regression as ways to deal with multicollinearity. It would be good if these strategies can be implemented into the algorithms. Even then, it is hard to justify that the hundreds of covariates are missing at random (MAR) *and* are related simply by sequences of linear and logistic regression models (MICE) or a single high-dimensional multivariate normal distribution (AMELIA). With GUIDE, there is no worry about collinearity nor with selecting the variables and interactions to include in the model.

Little (2020) noted that a missing value may refer to "not applicable" rather than a nonresponse. In the CE data, many variables with missing values have accompanying *missing-value flag* variables that give the reason for the missingness. Flag variables in the CE data typically have underscores in their names (e.g., the flag variable associated with `INTRDVX` is `INTRDVX_`). Each flag variable can take four possible values: `A` for "valid nonresponse or where a response in not anticipated", `C` for "don't know or refuse to answer", `D` for "valid data type", and `T` for "top-coded", where a response exceeding the 97th percentile is replaced by a constant that preserves the overall sample mean. Variable values associated with `A` and `C` flags are unobserved and are recorded as `NA`.

In the CE data, variable `AGE2` refers to the age of the respondent's spouse. About 40 percent of the values of `AGE2` = `NA`. That is due to many respondents being unmarried, widowed or divorced respondents, for whom `AGE2_` = `A`. The standard "separate models" approach calls for partitioning the data into two sets, fitting one model to the set where `AGE2_` = `A` with `AGE2` excluded, and another model to the other set with missing values in `AGE2` imputed. This approach is applicable only if missing-value flag variables are available. Even though they are in the CE data, the approach is impractical because there are 120 variables with missing-value flag variables, which could generate up to 2^{120} separate models.

For GUIDE, a split on `AGE2` takes the form "$\texttt{AGE2} \leq c$" or "$\{\texttt{AGE2} \leq c\} \cup \{\texttt{AGE2} = \texttt{NA}\}$". (The second split is shown in tree diagrams as "$\texttt{AGE2} \leq_* c$", where the notation "$X \leq_* c$" is an abbreviation for "$X \leq c$ or $X = \texttt{NA}$".) In LECL, a node can be split on `AGE2` only or on `AGE2_` only, but not on both. This restriction is removed in the current version of GUIDE, where a split can employ a variable and its flag simultaneously. This is shown in Figure 1, which is a GUIDE classification tree for predicting whether `INTRDVX_` = `C` (`INTRDVX` is missing) or `D` (`INTRDVX` is nonmissing). Table 1 gives the names, definitions and numbers of missing values of the variables in the figure. Splits at nodes 2, 16, 17, 18, 35, and 36 have the form "$\{X \leq c\}$ or $X_{\_} = \texttt{A}$", where $X_{\_}$ denotes the flag variable for X. Node 9 is split on "$\texttt{FEDRFNDX} \leq_* 260$", where all missing values, irrespective of type, go to the left branch. Node 141 is split solely on a missing-value flag variable, "`LIQU_YRX` = `A`."

A piecewise-constant estimate of the propensity that `INTRDVX` is nonmissing can be obtained from the proportions of observations with `INTRDVX_` = `D` in the terminal nodes of the classification tree. The propensity scores can then be used in an IPW estimate of the population mean of `INTRDVX`. Logistic regression is traditionally used to estimate propensity scores, but this requires imputation of the missing covariate values which is impossible to do here with MICE or AMELIA, due to the large numbers of covariates and large amount of missing

Table 1. Names, definitions and numbers of missing values of variables in Figure 1.

Name	Definition	#Missing
BATHRMQ	Number of complete bathrooms	21
BATHRMQ_	Flag variable for BATHRMQ	
EDUC_REF	Education of reference person	
FEDRFNDX	Federal income tax refund to all CU members	2,530
FEDR_NDX	Flag variable for FEDRFNDX	
FEDTAXX	Federal income tax paid by all CU members	3,752
FEDTAXX_	Flag variable for FEDTAXX	
FSALARYX	Wage and salary income of all CU members in past 12 months	
INTRDVX	Interest or dividend received in past 12 months	1,771
INTRDVX_	Flag variable for INTRDVX	
LIQUIDX	Value of checking, savings, CD, etc., accounts	3,827
LIQUIDX_	Flag variable for LIQUIDX	
LIQUDYRX	Total value of bank accounts one year ago	3,876
LIQU_YRX	Flag variable for LIQUDYRX	
OCCUCOD2	Highest paid job of spouse in last 12 months	2,832
OCCU_OD2	Flag variable for OCCUCOD2	
PSU	Primary sampling unit	
RESPSTAT	Completeness of income response (1=complete; 2=incomplete)	
RETSURVX	Retirement, survivor, disability pensions in past 12 months	3,520
RETS_RVX	Flag variable for RETSURVX	
SLRFUNDX	State and local income tax refund received by all CU members	3,167
SLRF_NDX	Flag variable for SLRFUNDX	
STATE	State identifier	

values. This is where GUIDE has a major advantage. Further, GUIDE performs variable selection automatically, reducing the number of variables from more than 600 to a handful in the figure. Finally, it is clear from the tree structure that there are interactions among the covariates. Logistic regression would be hard pressed to identify the interactions, even without missing values.

4. Bias

Multiple imputation was conceived years ago to reconstruct missing values in public-use data settings where the imputer and the analyst (the one analyzing the imputed data) were distinct entities and the objective was to make valid statistical statements about the analyst's linear model parameters. It has since been established that if the imputer's model is incompatible with the analyst's

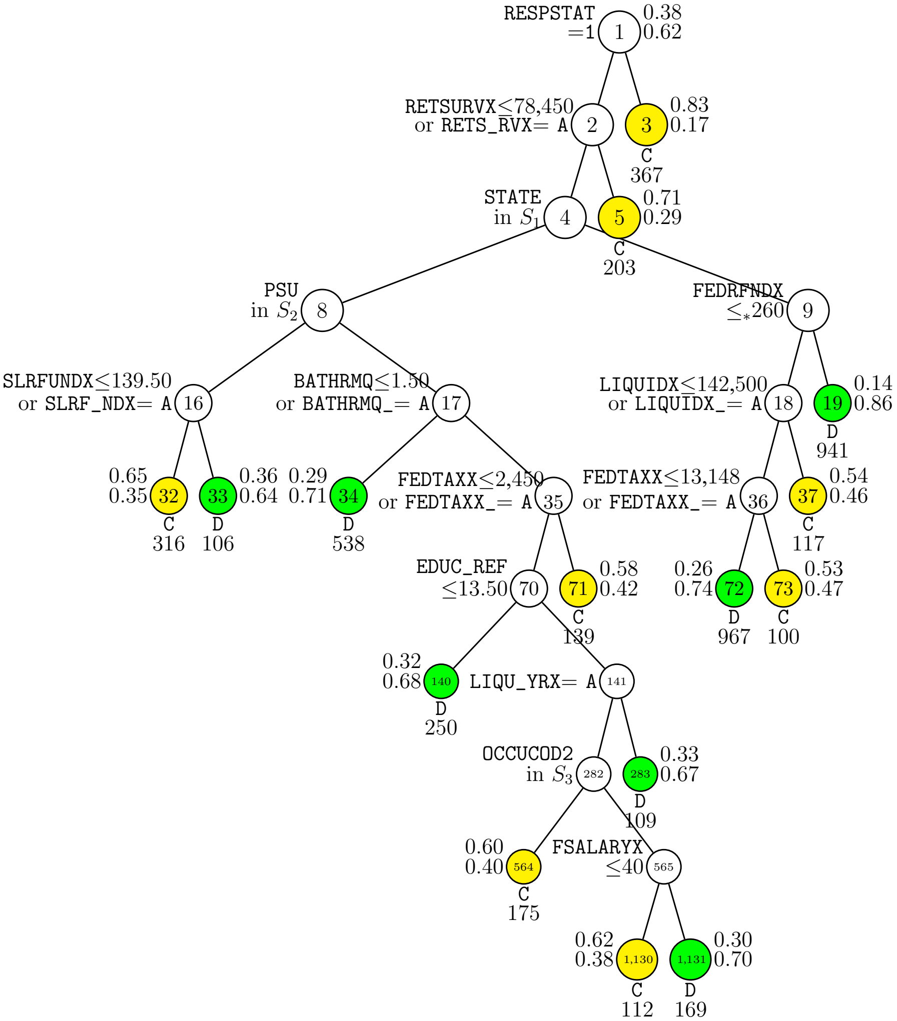

Figure 1. GUIDE classification tree for predicting INTRDVX_. At each split, an observation goes to the left branch if and only if the condition is satisfied. The symbol '$\le_*$' stands for '≤ or missing'. Set $S_1 = \{8, 9, 10, 12, 15, 17, 25, 26, 34, 36, 39, 42, 45, 47, 53, 55\}$; set $S_2 = \{1{,}102, 1{,}109, 1{,}111, 1{,}423\}$; and set $S_3 = \{2, 3, 5, 6, 10\}$. Predicted classes and sample sizes are printed below terminal nodes (yellow color for C and green for D). Estimated class posterior probabilities for INTRDVX_ = C and D are given beside the nodes.

model, the inferences may be invalid (Fay (1992); Rubin (1996)).

Although it does not require imputation for tree construction, GUIDE *can* be used to impute missing values as an alternative to AMELIA and MICE for other applications. Simply fit a GUIDE model to each variable as dependent variable in turn (without imputation of missing covariate values) and use the fitted values as imputed values. This one-variable-at-a-time approach is similar to MICE, except that MICE needs iteration because it first imputes all missing values with initial values and then fits a model to each variable in turn to the imputed data. Iteration is needed to reduce the effects of initial values and the imputation order of the variables, but iteration also propagates imputation errors. As an imputation method, GUIDE requires neither initialization nor iteration and is invariant of the order that variables are imputed.

We use five examples to compare estimation bias in linear models fitted to data imputed by GUIDE, MICE and its alternative IVEware (Raghunathan et al. (2016)) that Little (2020) favored. The examples span the range from correct model and missingness assumptions for MICE and IVEware to situations where they are violated. Two GUIDE methods are employed. The first is `gforest`, a GUIDE forest of 500 GUIDE piecewise-constant regression trees. It is similar to Breiman's random forest (Breiman (2001)), except that the latter uses CART trees, which have variable selection bias (Loh and Shih (1997); Loh (2002)). The second is `gstep`, a GUIDE regression tree with a stepwise linear regression model fitted in each node. Owing to its greater flexibility, `gstep` often has better prediction accuracy than a piecewise-constant tree, but it employs *local* imputation of missing values in each node with node means to fit the stepwise linear models. See LECL and Loh (2012, 2014) for comparisons between piecewise-constant trees, `gstep`, `gforest`, and random forest. Let $\mathrm{N}(0, \sigma^2)$ denote the normal distribution with mean 0 and variance σ^2 and $\mathrm{MVN}(\mu, \rho)$ the multivariate normal distribution with mean vector μ and covariance matrix with 1 along the diagonal and ρ in the off-diagonal elements. The following results are based on 1,000 simulation trials.

Example 1. Let $(X_1, X_2, \ldots, X_9)$ be $\mathrm{MVN}(0, 0.5)$ and $Y = X_1 + X_2 + \epsilon$, where ϵ is $\mathrm{N}(0, 0.1^2)$. Observations are MAR depending on X_9 such that given $X_9 = x$, the other variables (including Y) are independently missing with probability

$$p(x) = \frac{\exp(x - 1.8)}{1 + \exp(x - 1.8)}. \tag{4.1}$$

This results in about 20 percent missing values overall. Suppose that the analyst fits the model $EY = b_0 + \sum_{i=1}^{9} b_i X_i$ to a training sample of size n after imputation by MICE, `gforest`, or `gstep`. Figure 2 shows the estimated bias of the estimates of b_0, b_1 and b_2 for $n = 100$, 400 and 1,600. MICE and IVEware clearly deliver unbiased estimates throughout, as expected. The GUIDE methods are

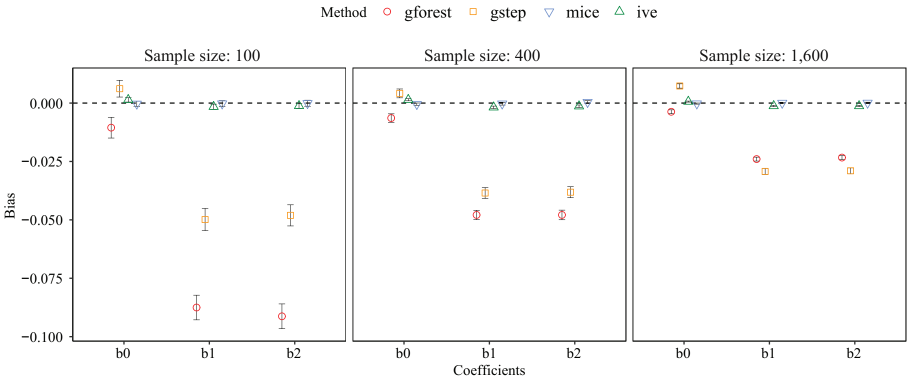

Figure 2. Estimated bias (with 2-SE bars) of regression coefficients using three imputation methods for Example 1. True model is $Y = X_1 + X_2 + \epsilon$ with $\epsilon \sim \mathrm{N}(0, 0.1^2)$ and $(X_1, X_2, \ldots, X_9)$ is multivariate normal with unit variance and common correlation 0.50. Values in $(Y, X_1, X_2, \ldots, X_8)$ are MAR depending on X_9. Fitted model is $EY = b_0 + \sum_{i=1}^{9} b_i X_i$.

not unbiased, but their biases tend to decrease (with `gforest` decreasing faster) as n increases.

Example 2. Variables $X_1, X_2, \ldots, X_9$, and ϵ are the same as in Example 1, but the true model is $Y = X_1^2 + \epsilon$ and the analyst's model is $EY = b_0 + b_1 X_1 + b_2 X_1^2$. The data are MAR depending on X_9 as before. Figure 3 shows that all three methods give biased estimates but the biases of the GUIDE methods are least.

Example 3. Let $(V_1, V_2, \ldots, V_9)$ be MVN$(0, 0.5)$ and $X_i = \Phi(V_i)$, where $\Phi(.)$ is the standard normal distribution function. The true model is $Y = X_1^2 + \epsilon$, where ϵ is independent $\mathrm{N}(0, 0.1^2)$ and the analyst's model is $EY = b_0 + b_1 X_1$. The data are MAR depending on X_9 with the missingness probability given $X_9 = x$ being $p(x) = 0.4x$. This mimics a case of model misspecification or one where the analyst wants to estimate a linear trend despite the true model being quadratic. The estimands are the means of the least squares estimates of the analyst's model for data without missing values $(b_0, b_1) = (-1/6, 1)$. The results in Figure 4 show that the bias of `gstep` is smallest and that of `gforest` and MICE are about equally large.

Example 4. Let $q_1 < q_2 < q_3$ denote the quartiles of the standard normal distribution. Variables $(X_1, X_6, X_7, X_8, X_9)$ are MVN$(0, 0.5)$ and $X_2 = X_1^2$, $X_3 =$

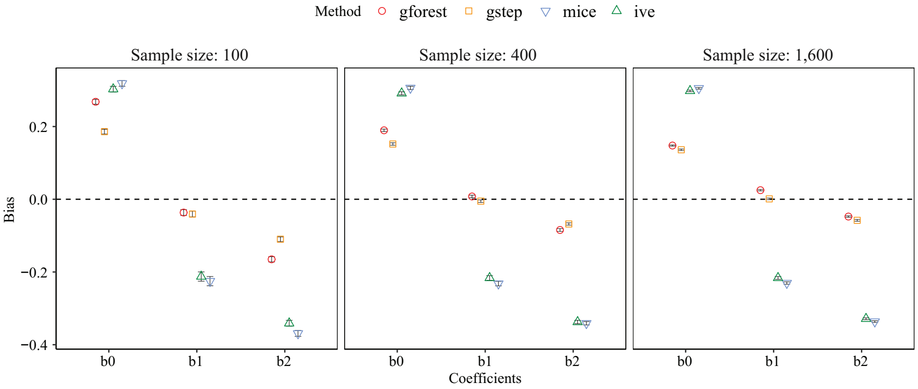

Figure 3. Estimated bias (with 2-SE bars) of regression coefficients using three imputation methods for Example 2. True model is $Y = X_1^2 + \epsilon$ with $\epsilon \sim \mathrm{N}(0, 0.1^2)$ and $(X_1, X_2, \ldots, X_9)$ is multivariate normal with unit variance and common correlation 0.50. Values in $(Y, X_1, X_2, \ldots, X_8)$ are MAR depending on X_9. Fitted model is $EY = b_0 + b_1 X_1 + b_2 X_1^2$.

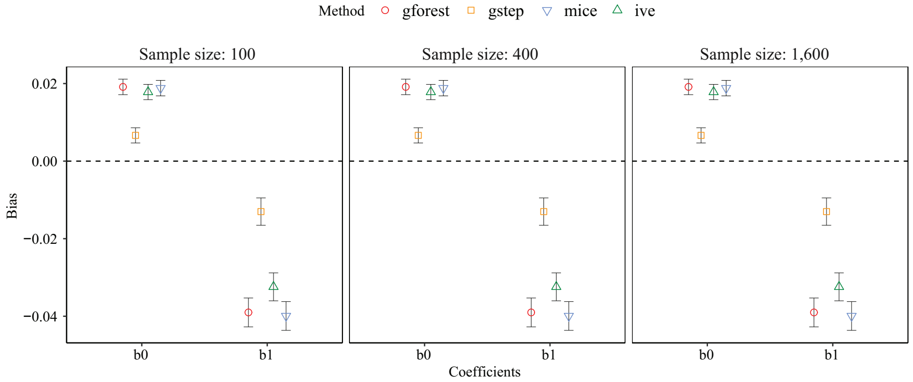

Figure 4. Estimated bias (with 2-SE bars) of regression coefficients using three imputation methods for Example 3. True model is $Y = X_1^2 + \epsilon$ with $\epsilon \sim \mathrm{N}(0, 0.1^2)$ and $(X_1, X_2, \ldots, X_9)$ is multivariate correlated uniform on nine-dimensional unit cube. Values in $(Y, X_1, X_2, \ldots, X_8)$ are MAR depending on X_9. Fitted model is $EY = b_0 + b_1 X_1$.

$1 + \sum_{j=1}^{3} I(X_1 \geq q_j)$, $X_4 = \sin(2\pi X_1)$, and $X_5 = \exp(X_1)$. The true model is $Y = X_1 + X_2 + X_3 + \epsilon$ with ϵ independent $\mathrm{N}(0, 0.1^2)$. The analyst's model is $EY = b_0 + \sum_{i=1}^{9} b_i X_i$. Variable X_1 is MNAR such that, given $X_1 = x$, it is missing with probability $p_1(x) = 0.8I(x < q_3) + 0.05I(x \geq q_3)$. Similarly, X_2 is MNAR such that given $X_2 = x$, it is missing with probability $p_2(x) = 0.8I(x < r_1) + 0.05I(x \geq r_1)$, where r_1 is the first quartile of the chi-squared

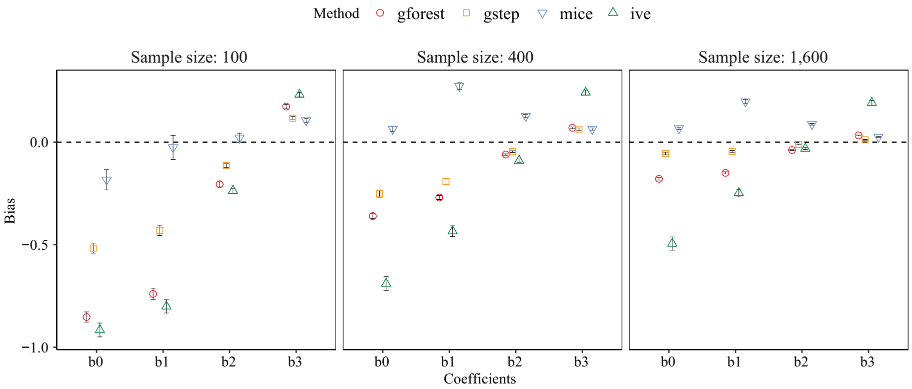

Figure 5. Estimated bias (with 2-SE bars) of regression coefficients using three imputation methods for Example 4. True model is $Y = X_1 + X_2 + X_3 + \epsilon$ and the fitted model is $EY = b_0 + \sum_{i=1}^{9} b_i X_i$, with X_1 and X_2 MNAR.

distribution with 1 degree of freedom. The results in Figure 5 show that MICE has the smallest bias for $n = 100$, but `gstep` is best for $n =$ 1,600. IVEware is consistently worst for all three sample sizes.

Example 5. The settings are the same as in Example 4, except that Y is MNAR, with probability of missing, $p(y) = 0.8I(Y < 1.75) + 0.05I(Y \geq 1.75)$. The value 1.75 is approximately the first quartile of the distribution of Y. Figure 6 shows that for $n = 400$ and 1,600, `gforest` has the smallest bias and IVEware the largest; no clear patterns are discernible for $n = 100$.

These results show that the strengths of MICE and IVEware are also their weaknesses. If the assumptions of the model and MAR are satisfied, they are unbiased. Otherwise, `gstep` and `gforest` may have smaller bias.

5. Prediction

Little (2020) stated that "the impression given by LECL is that the tree algorithm will automatically lead to good predictions of missing values. I think this uncritical assessment is common for algorithmic methods, where the underlying model is treated as a 'black box' and not explicitly scrutinized." In a large study of real data without missing values, Lim, Loh and Shih (2000) showed that there is much variation in the prediction accuracy of model-based and algorithmic classification methods. We complement those results here with a simulation study of 2 model-based and 7 machine learning regression methods when they

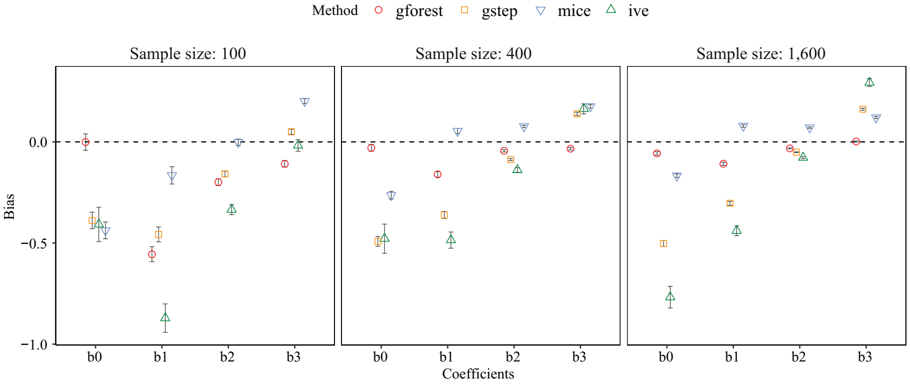

Figure 6. Estimated bias (with 2-SE bars) of regression coefficients using three imputation methods for Example 5. True model is $Y = X_1 + X_2 + X_3 + \epsilon$ and the fitted model is $EY = b_0 + \sum_{i=1}^{9} b_i X_i$, with X_1, X_2 and Y MNAR.

are applied to data after missing-value imputation. We employed 24 real data sets originally without missing values (see the Supplementary Materials section for details of the data sets). Each data set was randomly split into a training set and a test set in a 10-fold cross-validation fashion explained in Section 5.4 below. Values in the predictor variables were made randomly missing according to MCAR and MAR mechanisms. GUIDE forest, MICE, and mean/mode (MM) imputation were used to impute the missing values. Then each regression method was fitted to the imputed training data and the accuracy of its predicted values assessed with the test set. Sections 5.1 and 5.2 present the regression methods and imputation methods, respectively. Section 5.3 describes the missing-data mechanisms and Section 5.4 gives the results.

5.1. Nine regression methods

SLR. Stepwise linear regression using the R functions `lm` and `step` (R Core Team (2016)).

LASSO. Regularized linear regression in `glmnet` package with tuning parameter λ selected by cross-validation and the 1-SE rule (Friedman, Hastie and Tibshirani (2010)).

MARS. Multivariate adaptive regression splines in `mars` function of `mda` package (Hastie et al. (2016)).

SVR. Support vector machine regression in `e1071` R package (Meyer et al.

(2015)).

RPART. Regression tree in `rpart` package (Therneau, Atkinson and Ripley (2018)), an implementation of CART (Breiman et al. (1984)) that fits a piecewise-constant regression tree model with the 0-SE tree pruning rule.

GLT. The `gstep` GUIDE method in the previous section.

M5. Piecewise-linear regression tree (Quinlan (1992)) from the `RWeka` package (Hornik, Buchta and Zeileis (2009)). It constructs a piecewise-constant tree first and then fits the data in each terminal node with a multiple linear regression model.

RF. Random forest (Breiman (2001)) in the `RandomForest` package (Liaw and Wiener (2002)).

GF. The `gforest` GUIDE method of the previous section.

5.2. Imputation methods

Mean/Mode imputation (MM). For training data imputation, all missing values in each variable were replaced by their sample means (for ordinal variables) or sample modes (for categorical variables). The same training-sample means and modes were also used to impute missing values in the test data.

MICE (MI). This uses the `mice` package (van Buuren and Groothuis-Oudshoorn (2011)), which creates 5 imputed data sets from the training sample. Each prediction method was applied to the imputed data sets to produce 5 predicted values for each test observation which were then averaged to yield a final predicted value. Missing values in the test sample were imputed by the training-sample means and modes.

GUIDE forest (GI). This fits a GUIDE forest to each variable as dependent variable in the training set, using the other variables as predictor variables. The fitted forest model is used to impute missing values in the respective variable in the training and test data sets. Because the forest is an ensemble of piecewise-constant GUIDE trees, it does not require missing values in predictor variables to be imputed. Therefore, unlike MICE, GI does not require iteration and the variables can be imputed in any order.

Algorithm-specific defaults (DF). As stand-alone methods, GF, GLT, M5 and RPART each has its own method of dealing with missing values. GF does not

Table 2. Prediction-imputation methods grouped by imputation technique.

Name	Description
Algorithm-default techniques for missing values	
GLT_DF	GUIDE stepwise regression tree without imputation
GF_DF	GUIDE regression forest without imputation
M5_DF	M5 with case weights
RPART_DF	RPART with surrogate splits
Imputation by training sample means and modes (MM)	
GLT_MM	GUIDE stepwise regression tree with mean/mode imputation
GF_MM	GUIDE regression forest with mean/mode imputation
LASSO_MM	LASSO with mean/mode imputation
M5_MM	M5 with mean/mode imputation
MARS_MM	MARS with mean/mode imputation
RF_MM	Random forest with mean/mode imputation
RPART_MM	RPART with mean/mode imputation
SLR_MM	Stepwise linear regression with mean/mode imputation
SVR_MM	Support vector regression with mean/mode imputation
Imputation by GUIDE forest (GI)	
GLT_GI	GUIDE stepwise regression tree with GUIDE forest imputation
GF_GI	GUIDE regression forest with GUIDE forest imputation
LASSO_GI	LASSO with GUIDE forest imputation
M5_GI	M5 with GUIDE forest imputation
MARS_GI	MARS with GUIDE forest imputation
RF_GI	Random forest with GUIDE forest imputation
RPART_GI	RPART with GUIDE forest imputation
SLR_GI	Stepwise linear regression with GUIDE forest imputation
SVR_GI	Support vector regression with GUIDE forest imputation
Imputation of training samples by MICE, test samples by training-sample means and modes	
GLT_MI	GUIDE stepwise regression tree with MICE imputation
GF_MI	GUIDE regression forest with MICE imputation
LASSO_MI	LASSO with MICE imputation
M5_MI	M5 with MICE imputation
MARS_MI	MARS with MICE imputation
RF_MI	Random forest with MICE imputation
RPART_MI	RPART with MICE imputation
SLR_MI	Stepwise linear regression with MICE imputation
SVR_MI	Support vector regression with MICE imputation

impute because it is a forest of 500 piecewise-constant GUIDE trees. GLT uses the node training-sample means to impute missing missing values prior to fitting the linear regression model in the node. In M5, an observation with missing value in the split variable is randomly sent to one of the child nodes with probability proportional to its sample size. RPART uses surrogate splits (Breiman et al. (1984)).

Table 2 lists the 31 procedures obtained by combining the 9 prediction algorithms with the 4 missing value techniques (some algorithms do not have DF techniques). To aid recall, the name of each procedure consists of two parts joined by an underscore. The first part is the name of the prediction method and the second part is the imputation method. For example, GLT_DF, GF_DF, M5_DF and RPART_DF employ the algorithm-specific missing value techniques of GLT, GF, M5 and RPART, respectively, and GLT_MM applies GLT to data imputed by the MM method.

5.3. Missing-data mechanisms

Let the vector of response and predictor variables be $(Y, X_1, X_2, \ldots, X_K)$ and let the ith observation vector be $(Y_i, X_{i1}, X_{i2}, \ldots, X_{iK})$. Let $p = 0.05$ or 0.20 denote the proportion of missing values in a data set. Three ways to create missing values among the X variables were studied. The first is MCAR, the second makes each X_{ij} value missing with probability depending on the value of Y_i, and the third makes X_{ij} missing depending on the value of X_{ik} for some $k \neq j$.

Missing completely at random (MCAR). Each X_{ij} is independently missing with probability p.

Missing at random depending on Y (MAR_y). Here the value of X_{ij} is more likely to be missing if the value of Y_i is large. Let S denote the set of indices i such that Y_i is greater than its third quartile. For each (i, j), X_{ij} is independently missing with probability $8p/5$ if $i \in S$ and with probability $4p/5$ if $i \notin S$.

Missing at random depending on X (MAR_x). Following Twala (2009), missing values in X variables are randomly generated based on the values of other correlated variables. The procedure is as follows.

1. If a variable X is ordinal, turn it into a 4-level categorical variable by discretization at the sample quartiles.
2. Perform a chi-squared test of independence for each pair of (categorical or discretized ordinal) X variables.
3. Order the pairs from smallest to largest according to p-value, omitting pairs where at least one variable in the pair is a member of another pair with a smaller p-value.
4. For each pair that remains:

(a) Randomly select one variable from the pair; call it U and the other V.

(b) Randomly select one value of U and call it c.

(c) Let (U_i, V_i) denote the values of (U, V) for observation i. If $U_i = c$, make V_i and its corresponding X value missing with probability $2q$. Otherwise if $U_i \neq c$, make V_i and its corresponding X value missing with probability q. The value of q is chosen so that the overall proportion of missing X values in the data set is p.

5.4. Results

The prediction accuracy of the methods was measured by relative mean square error (rMSE). For each data set, missing values were created using one of the above missing data generation mechanisms; then the following ten-fold cross-validation procedure was used to estimate the rMSE of each method.

1. Randomly divide the data set L into ten disjoint subsets $L_1, \ldots, L_{10}$, with each containing approximately the same number of records.

2. For $k = 1, \ldots, 10$, let L_k be the test set, and $L - L_k$ be the training set. If required, impute missing values in the training and test sets with the given imputation method.

3. Fit a prediction model to each training set and use it to predict the y_i values in L_k. Let the predicted value be denoted by $\hat{y}_i^{(k)}$ and let the mean of the response values in $L - L_k$ be denoted by $\bar{y}^{(k)}$. The relative mean square error is

$$\text{rMSE} = 10^{-1} \sum_{k=1}^{10} \frac{\sum_{i \in L_k} \left(y_i - \hat{y}_i^{(k)} \right)^2}{\sum_{i \in L_k} \left(y_i - \bar{y}^{(k)} \right)^2}$$

where smaller values indicate higher accuracy.

The prediction method types are grouped by color in the graphs below, with blue for regression trees (GLT, M5 and RPART), red for regression forests (GF and RF), green for linear models (LASSO and SLR), and yellow for MARS and SVR.

5.4.1. MCAR

Figure 7 shows the mean rMSE over the 24 data sets of each method under MCAR. For 5% missing, the mean rMSEs range from 0.50 to less than 0.80. The best methods are M5_MI and GLT_MI, which are both tree methods applied to

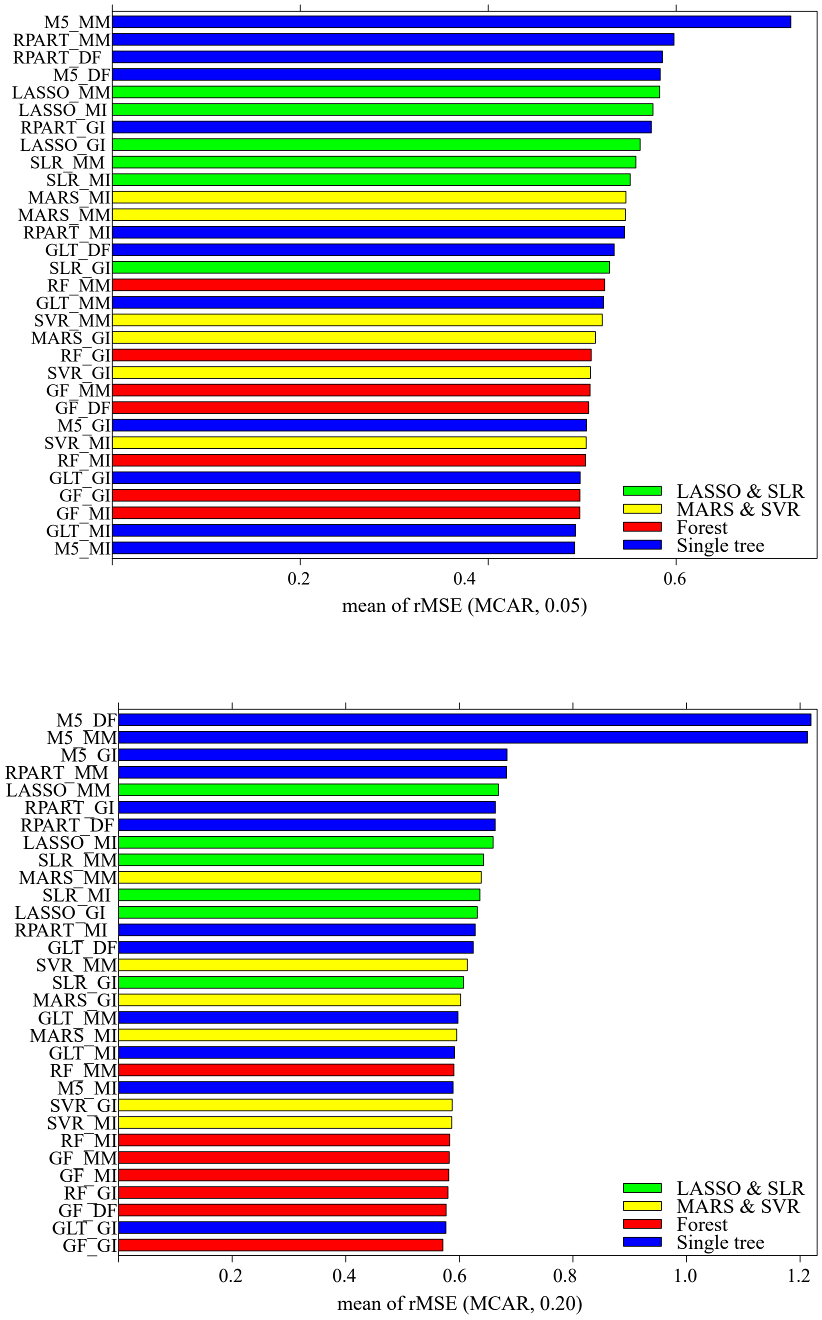

Figure 7. Mean rMSE over 24 data sets for 5% (top) and 20% (bottom) MCAR. Shorter bars are better.

training data imputed with MICE and test data imputed with training sample means and modes. They are followed closely by GUIDE forest (GF_MI and GF_GI) and the tree method GLT_GI. Across all imputation techniques, the mean rMSEs of the GF methods are almost indistinguishable from that of M5_MI and GLT_MI. RF has higher mean rMSE than GF, irrespective of imputation method. MARS and SVR fall in the middle third among the nine regression methods. LASSO and SLR are below average, for all imputation methods. M5_MM is the worst by a wide margin. Given that M5_MI is best, this shows that (for M5 at least) imputation method can have a large effect on the performance of a regression method.

For 20% missing, the mean rMSEs of all but two methods are close to each other, the exceptions being M5_MM and M5_DF, whose mean rMSEs are about twice as large as the rest. M5_GI is third last. Regression forests (GF and RF) dominate, with imputation method making little difference. The only non-forest algorithm to break into the top is GLT_GI which is second best. RF is slightly inferior to GF, for every imputation method. Again, MARS and SVR are middling and LASSO and SLR are worse, for all imputation methods.

The results suggest that the best methods for both 5% and 20% MCAR missing are GF_GI, GF_MI and GLT_GI. Methods M5_MM and M5_DF are consistently among the worst; LASSO and SLR are below average; and MARS and SVR are middling.

5.4.2. MAR_y

Figure 8 shows the corresponding results for MAR_y. They are similar to the case for MCAR in that (i) forest methods (GF and RF) are among the best, irrespective of imputation method, (ii) single-tree GLT is as good as the best for 5% missing, (iii) MARS and SVR are middling, (iv) LASSO and SLR are below average, and (v) M5 is worst, by a large margin for 20% missing, irrespective of imputation method.

5.4.3. MAR_x

Figure 9 shows the results for MAR_x missing. For 5% missing, the single tree GLT_GI is the best (same as for MAR_y missing). GLT with other imputation methods and forest methods (for all imputation methods) are close behind. Again MARS and SVR are middling and LASSO and SLR are below average. M5_DF is worst. For 20% missing, the results are roughly similar to those for MAR_y: many forest methods are best and M5_DF is worst.

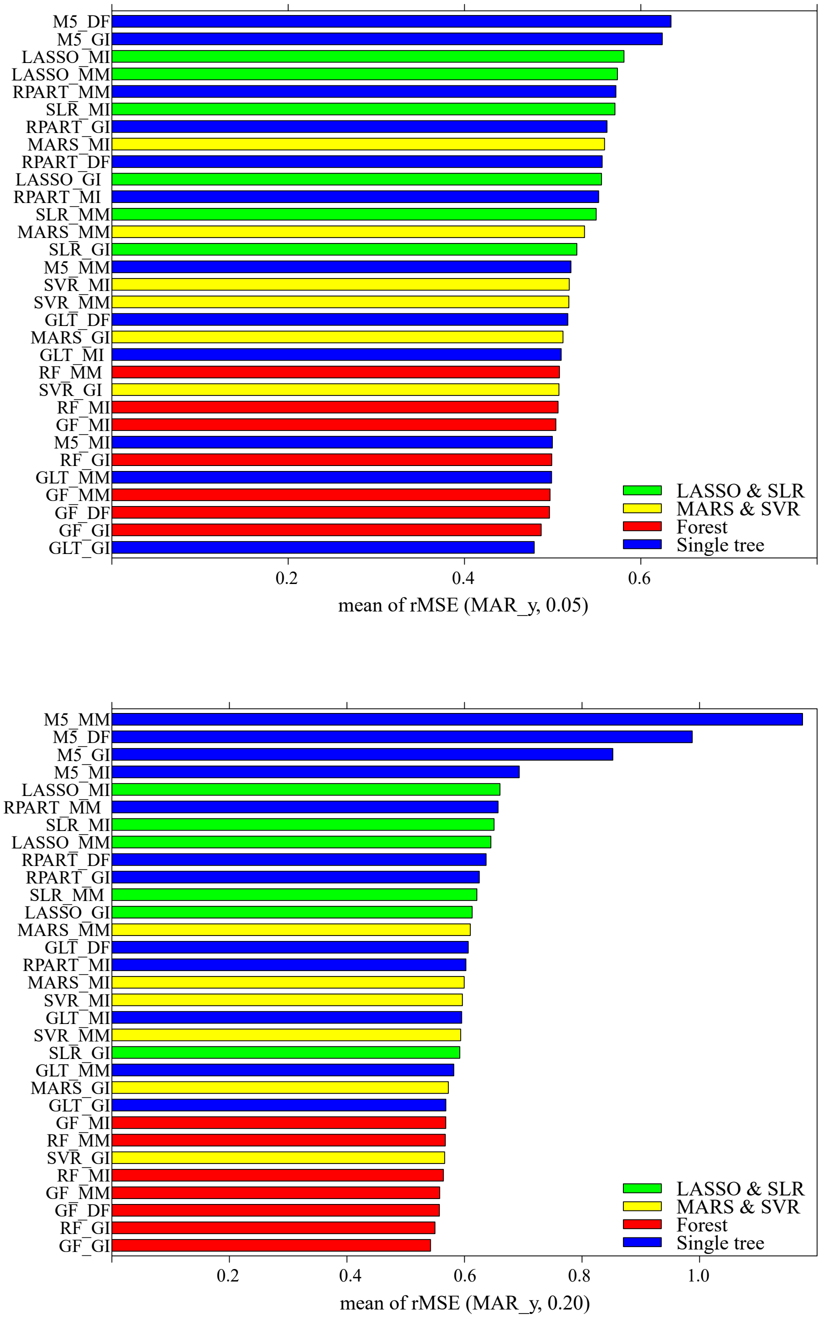

Figure 8. Mean rMSE over 24 data sets for 5% (top) and 20% (bottom) MAR_y. Shorter bars are better.

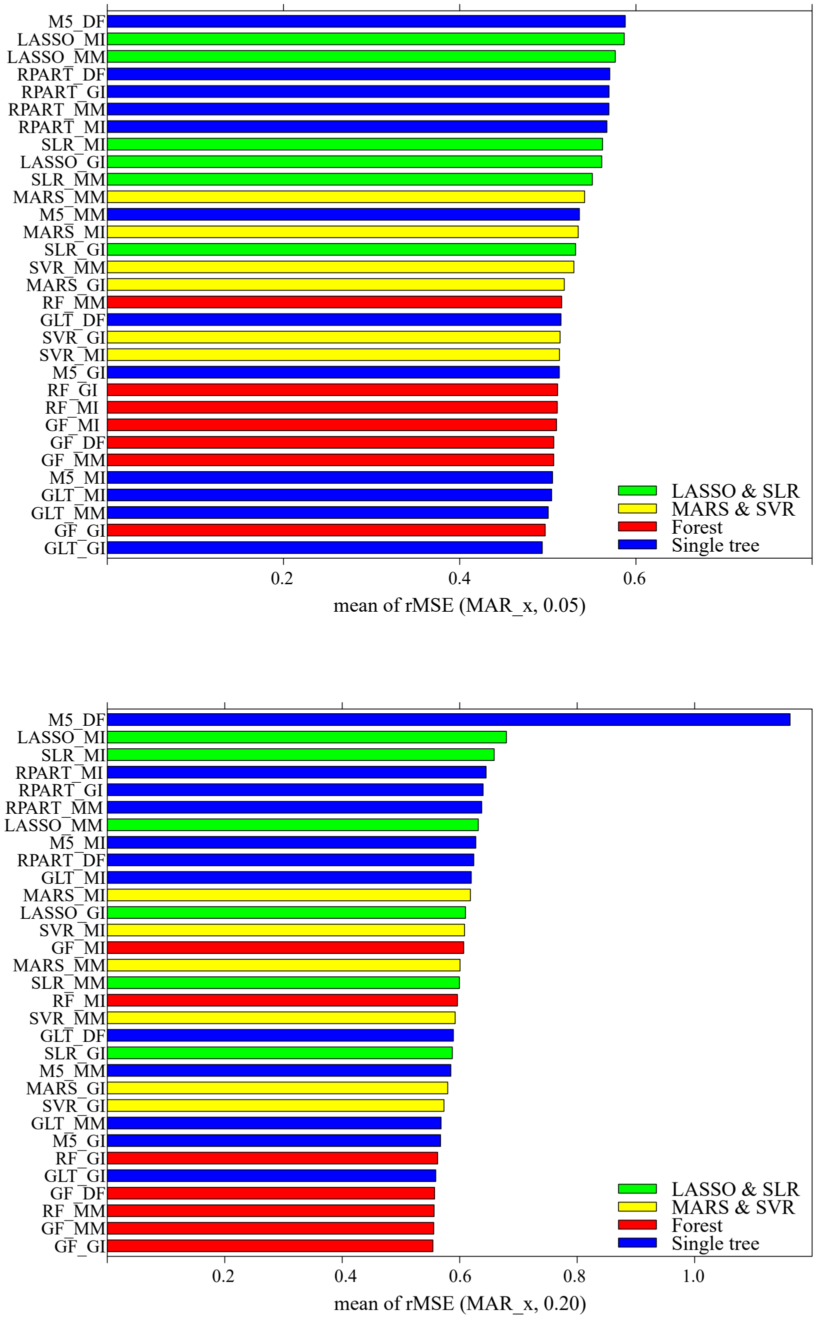

Figure 9. Mean rMSE over 24 data sets for 5% (top) and 20% (bottom) MAR_x. Shorter bars are better.

Table 3. Performance of methods. Columns (a), (b) and (c) refer to MCAR, MAR_y and MAR_x. A check mark (✓) indicates the mean rMSE of a method ranks in the top half of methods; a dash (−) indicates no default (DF) imputation method.

Miss.			DF			GI			MI			MM	
rate		(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
5%	GLT		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	M5				✓		✓	✓	✓	✓			
	RPART												
	GF	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	RF	−	−	−	✓	✓	✓	✓	✓	✓		✓	✓
	MARS	−	−	−	✓	✓							
	SVR	−	−	−	✓	✓	✓	✓		✓	✓	✓	
	LASSO	−	−	−									
	SLR	−	−	−									
20%	GLT				✓	✓	✓	✓	✓	✓	✓	✓	✓
	M5							✓					
	RPART												
	GF	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	RF	−	−	−	✓	✓	✓	✓	✓	✓		✓	✓
	MARS	−	−	−	✓	✓	✓	✓					
	SVR	−	−	−	✓	✓	✓	✓	✓	✓		✓	✓
	LASSO	−	−	−									
	SLR	−	−	−		✓	✓						

5.4.4. Summary

Overall across all three missing-value mechanisms, GF_GI is best when the missing rate is 20 percent. The closeness of the mean rMSEs of other methods indicate, however, that there are other good methods too. To highlight this, we classify each method as "good" or "bad" according to whether its mean rMSE is below or above the median of all methods, for each missing rate and each missing-value mechanism. Table 3 gives the results of the classification. A "good" prediction-imputation method is shown with a checkmark (✓). Methods that do not have their own default missing value handling techniques are marked with a dash (−). Columns labeled (a), (b), and (c) refer to the MCAR, MAR_y and MAR_x mechanisms. Rows with the most checkmarks identify the best regression methods. Similarly, columns with the most checkmarks identify the best imputation method. By these criteria, GF is the best regression method, followed by RF and GLT. The worst methods are LASSO and RPART, which never place in the top half of all methods. SLR is slightly better, but only if it uses GI imputation. Among imputation methods, GI is the best, with MI second and MM last.

6. Conclusion

There are many misconceptions about machine learning methods in general and regression tree methods in particular. Some people think that because the methods are not model-based, they are "black boxes" and hence cannot be trusted. The reason the methods are not model-based is because they are designed to be free of model assumptions. The GUIDE piecewise-constant tree and GUIDE forest are two prime examples. Because neither one makes any assumptions on the form of the true regression function, their predicted values tend to be more accurate over a larger class of functions (including non-smooth and discontinuous functions) than model-based methods. The keys to their prediction accuracy, besides effective recursive partitioning, are tree pruning by cross-validation for GUIDE tree and model averaging for GUIDE forest. As explained in Section 3, GUIDE has an additional advantage if there are missing data, because it uses missing covariate values "as is" without imputation at each split. Although this design decision was made years ago as an alternative to surrogate splits, it has advantages to imputation techniques such as MICE that require MAR assumptions *for all predictor variables.*

To show that there are statistical benefits as well when there are missing values, we compared GUIDE with MICE and other methods on two performance criteria. The first criterion is estimation bias in regression coefficients and the second is prediction accuracy after missing value imputation. Traditional model-based imputation methods were invented to enable statistical inference on parameters in hypothesized data models; prediction was not the main focus. Not surprisingly, we found that if the model and MAR assumptions are satisfied, MICE yields unbiased estimates. Machine learning methods such as GUIDE, however, are designed for prediction. Because they do not make explicit assumptions about the true regression function or the missingness mechanisms, there is no reason to expect that they will produce unbiased regression coefficient estimates in linear models. Nevertheless, Section 4 shows that although GUIDE imputation methods do not yield unbiased estimates under conditions ideal for MICE, their biases seem to diminish with increasing sample size. But if the assumptions for MICE are violated, GUIDE can have less bias.

The second criterion of prediction accuracy after missing data imputation is harder to evaluate, because three factors are involved: the imputation method, the prediction model, and the type of missingness mechanism. We chose three imputation methods (MICE, GUIDE and mean/mode) and nine prediction models (including stepwise linear regression, LASSO, MARS, support vector regression, and trees and forests) and evaluated their combined performance on 24 real data sets with missing values simulated under three schemes (MCAR and MAR depending on X or Y). Section 5 shows that, averaged over imputation methods, the prediction accuracy of tree-based methods ranges from the best (GUIDE) to the worst (M5 and RPART) of the lot. MARS and SVR tend to be average and stepwise linear regression and LASSO are below average. Averaged over prediction methods and missingness schemes, GUIDE imputation was best, followed by MICE and mean/mode imputation.

The boundary between model-based and machine learning methods is blurred nowadays as statisticians and machine learners adopt each other's techniques. LASSO, for example, is based on the linear model, but it employs machine learning ideas such as cross-validation for tuning parameter selection. Conversely, the "model-based" MOB regression tree method (Zeileis, Hothorn and Hornik (2008)) is so named because it fits a parametric model in each node of the tree. It is ideal if every method can be analyzed theoretically, but such methods must necessarily be relatively simple because there are limits to mathematical analysis. This rules out most algorithmic methods, but it does not mean that they are not worthy of consideration. The good news is that many methods now come with software that allows their performance to be evaluated by simulation. Simulations are never definitive, but they can be as varied and realistic as desired. Furthermore, the simulations may be performed by anyone besides the developers. For example, Lee and Jeong (2017) reported that GUIDE imputation compared favorably against AutoImpute, a proprietary algorithm developed by Westat, and Liu et al. (2019), Loh, Cao and Zhou (2019) and Loh and Zhou (2020) demonstrated that GUIDE performs well against other methods for subgroup identification methods in precision medicine.

Supplementary Materials

Details about the data sets used in Section 5 are available online at the journal website.

Acknowledgments

The authors thank Yuanzhi Li and Xu Xu for contributing to the results in Section 5 and to Moon-Jung Cho for helpful comments. Loh's research was supported in part by grants from the National Science Foundation and the University of Wisconsin Graduate School.

References

Breiman, L. (2001). Random forests. *Machine Learning* **45**, 5–32.

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984). *Classification and Regression Trees.* Wadsworth, Belmont.

Cai, T., Cai, T. T. and Zhang, A. (2016). Structured matrix completion with applications to genomic data integration. *Journal of the American Statistical Association* **111**, 621–633.

Candès, E. J. and Recht, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational Mathematics* **9**, 717–772.

Chaudhuri, P., Huang, M.-C., Loh, W.-Y. and Yao, R. (1994). Piecewise-polynomial regression trees. *Statistica Sinica* **4**, 143–167.

Chaudhuri, P., Lo, W.-D., Loh, W.-Y. and Yang, C.-C. (1995). Generalized regression trees. *Statistica Sinica* **5**, 641–666.

Chaudhuri, P. and Loh, W.-Y. (2002). Nonparametric estimation of conditional quantiles using quantile regression trees. *Bernoulli* **8**, 561–576.

Chi, E. C., Zhou, H., Chen, G. K., Del Vecchyo, D. O. and Lange, K. (2013). Genotype imputation via matrix completion. *Genome Research* **23**, 509–518.

Fay, R. E. (1992). When are inferences from multiple imputation valid? In *Proceedings of the Survey Research Methods Section*, 227–232. American Statistical Association.

Friedman, J., Hastie, T. and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* **33**, 1–22.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *The Annals of Statistics* **19**, 1–67.

Hastie, T., Tibshirani, R., F. Leisch, K. H. and Ripley, B. D. (2016). *mda: Mixture and flexible discriminant analysis.* R package version 0.4-9.

Honaker, J., King, G. and Blackwell, M. (2011). Amelia II: A program for missing data. *Journal of Statistical Software* **45**, 1–47.

Hornik, K., Buchta, C. and Zeileis, A. (2009). Open-source machine learning: R meets weka. *Computational Statistics* **24**, 225–232.

Jiang, X., Zhang, L. and Qiao, L. (2018). Completing missing exam scores with structural information and beyond. *Journal of Applied Remote Sensing* **13**, 1–11.

Knol, M. J., Janssen, K. J. M., Donders, A. R. T., Egberts, A. C. G., Heerdink, E. R., Grobbee, D. E., Moons, K. G. M. and Geerlings, M. I. (2010). Unpredictable bias when using the missing indicator method or complete case analysis for missing confounder values: an empirical example. *Journal of Clinical Epidemiology.*

Koren, Y., Bell, R. and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer* **42**, 30–37.

Lee, H. and Jeong, D. (2017). Missing data imputation using regression and classification tree software GUIDE. In *JSM Proceedings.* Survey Research Methods Section, American Statistical Association.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News* **2**, 18–22.

Lim, T.-S., Loh, W.-Y. and Shih, Y.-S. (2000). A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning Journal* **40**, 203–228.

Little, R. (2020). On algorithmic and modeling approaches to imputation in large data sets. *Statistica Sinica*, to appear.

Liu, Y., Ma, X., Zhang, D., Geng, L., Wang, X., Zheng, W. and Chen, M.-H. (2019). Look before you leap: systematic evaluation of tree-based statistical methods in subgroup identification. *Journal of Biopharmaceutical Statistics* **29**.

Loh, W.-Y. (2002). Regression trees with unbiased variable selection and interaction detection. *Statistica Sinica* **12**, 361–386.

Loh, W.-Y. (2009). Improving the precision of classification trees. *Annals of Applied Statistics* **3**, 1710–1737.

Loh, W.-Y. (2012). Variable selection for classification and regression in large p, small n problems. In *Probability Approximations and Beyond*, volume 205 of *Lecture Notes in Statistics—Proceedings* (Edited by Barbour, A., Chan, H. P. and Siegmund, D.), 133–157. Springer, New York.

Loh, W.-Y. (2014). Fifty years of classification and regression trees (with discussion). *International Statistical Review* **34**, 329–370.

Loh, W.-Y., Cao, L. and Zhou, P. (2019). Subgroup identification for precision medicine: a comparative review of thirteen methods. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **9**, e1326.

Loh, W.-Y., Eltinge, J., Cho, M. J. and Li, Y. (2019). Classification and regression trees and forests for incomplete data from sample surveys. *Statistica Sinica* **29**, 431–453.

Loh, W.-Y. and Shih, Y.-S. (1997). Split selection methods for classification trees. *Statistica Sinica* **7**, 815–840.

Loh, W.-Y. and Zhou, P. (2020). The GUIDE approach to subgroup identification. In *Design and analysis of Subgroups with Biopharmaceutical Applications* (Edited by Ting, N., Cappelleri, J. C., Ho, S. and Chen, D.-G.), 147–165. Springer, Nature Switzerland, Cham.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A. and Leisch, F. (2015). *e1071: Misc functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. R package version 1.6-7.

Quinlan, J. R. (1992). Learning with continuous classes. In *5th Australian Joint Conference on Artificial Intelligence*, 343–348. World Scientific, Singapore.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Raghunathan, T., Solenberger, P., Berglund, P. and Van Hoewyk, J. (2016). *IVEware: Imputation and Variance Estimation Software (version 0.3)*. University of Michigan.

Rubin, D. B. (1996). Multiple imputation after 18+ years. *Journal of the American Statistical Association* **91**, 473–489.

Therneau, T., Atkinson, B. and Ripley, B. (2018). *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1-13.

Tomasi, C. and Kanade, T. (1993). Shape and motion from image streams: a factorization method. *Proceedings of the National Academy of Sciences* **90**, 9795–9802.

Twala, B. (2009). An empirical comparison of techniques for handling incomplete data using decision trees. *Applied Artificial Intelligence* **23**, 373–405.

Vach, W. and Blettner, M. (1991). Biased estimation of the odds ratioin case-control studies due to the use of ad hoc methods of correcting for missing values for confounding variables. *American Journal of Epidemiology* **134**, 895–907.

van Buuren, S. (2012). *Flexible Imputation of Missing Data.* CRC Press, Boca Raton.

van Buuren, S. and Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software* **45**, 1–67.

Zeileis, A., Hothorn, T. and Hornik, K. (2008). Model-based recursive partitioning. *Journal of Computational and Graphical Statistics* **17**, 492–514.

Department of Statistics, University of Wisconsin, Madison, WI 53706, USA.

E-mail: loh@stat.wisc.edu

School of Mathematical and Statistical Sciences, Clemson University, Clemson, SC 29634, USA.

E-mail: qiongz@clemson.edu

Takeda Development Center Americas, Inc., Cambridge, MA 02139, USA.

E-mail: wenwen.zhang@takeda.com

Department of Statistics, University of Wisconsin, Madison, WI 53706, USA.

E-mail: pzhou9@wisc.edu

(Received March 2019; accepted June 2020)