Variable selection with missing data in both covariates and outcomes: Imputation and machine learning

Liangyuan Hu[1], **Jung-Yi Joyce Lin**[2], **Jiayi Ji**[2]

[1]Department of Biostatistics and Epidemiology, Rutgers University School of Public Health, USA

[2]Department of Population Health Science & Policy, Icahn School of Medicine at Mount Sinai, USA

Abstract

The missing data issue is ubiquitous in health studies. Variable selection in the presence of both missing covariates and outcomes is an important statistical research topic but has been less studied. Existing literature focuses on parametric regression techniques that provide direct parameter estimates of the regression model. In practice, parametric regression models are often sub-optimal for variable selection because they are susceptible to misspecification. Flexible nonparametric machine learning methods considerably mitigate the reliance on the parametric assumptions, but do not provide as naturally defined variable importance measure as the covariate effect native to parametric models. We investigate a general variable selection approach when both the covariates and outcomes can be missing at random and have general missing data patterns. This approach exploits the flexibility of machine learning modeling techniques and bootstrap imputation, which is amenable to nonparametric methods in which the covariate effects are not directly available. We conduct expansive simulations investigating the practical operating characteristics of the proposed variable selection approach, when combined with four tree-based machine learning methods, XGBoost, Random Forests, Bayesian Additive Regression Trees (BART) and Conditional Random Forests, and two commonly used parametric methods, lasso and backward stepwise selection. Numeric results suggest that when combined with bootstrap imputation, XGBoost and BART have the overall best variable selection performance with respect to the F_1 score and Type I error, while the lasso and backward stepwise selection have subpar performance across various settings. In general, there is no significant difference in the variable selection performance due to imputation methods. Guidance for choosing methods appropriate to the structure of the analysis data at hand are discussed. We further demonstrate the methods via a case study of risk factors for 3-year incidence of metabolic syndrome with data from the Study of Women's Health Across the Nation.

Keywords

Missing at random; Variable importance; Tree ensemble; Bootstrap imputation; Variable selection

Corresponding author: Liangyuan Hu, Department of Biostatistics and Epidemiology, Rutgers University School of Public Health, 683 Hoes Lane West, Piscataway, NJ, USA 08854. liangyuan.hu@rutgers.edu.

1 Introduction

Variable selection is an important statistical research topic concerning the identification of important predictors of an outcome variable. The building block of variable selection is the modeling of the covariate-outcome relationship. Traditionally, parametric models have been used to describe how the covariates are related to the outcome; and variables can be selected based on hypothesis tests between nested models or by optimizing a likelihood penalized for model complexity, typically using the Akaike or Bayesian information criterion. With parametric regression, the dependence structures among and distributional shapes of covariates need to be made explicit. Misspecification of these parametric forms will have an adverse effect on variable selection[1]. Considerable research has demonstrated advantages of variable selection based on flexible machine learning modeling techniques compared to methods relying on parametric regression[1–6]. In particular, nonparametric tree-based methods including Bayesian Additive Regression Trees (BART)[7], Random Forests (RF)[8] and extreme gradient boosting (XGBoost)[9] have attracted a lot of attention in the scientific literature. Tree-based machine learning methods mitigate the reliance on the parametric modeling assumptions and are more adept at capturing nonlinearities and interactions in the relationships among response and covariates[7,10,11], and have been widely applied in health research[2,12–17]. Variable selection procedures for these nonparametric tree-based methods typically rely on the variable importance measure supplied by the model. For example, permutation-based[1] or sequential stepwise approaches[18] have been used to select a set of important covariates that would optimize a performance metric, e.g., classification accuracy for a binary outcome or the mean-squared-error for a continuous outcome. These procedures can better represent complex response surfaces but are "black-boxes" in the sense that the effects of covaraites on the outcome are not directly quantified[1].

Missing data are ubiquitous in health care databases, and present substantial challenges to variable selection, as variable selection procedures need to be tailored to various missing data mechanisms and statistical approaches used for handling missing data under the specific mechanisms. There are three general missing data mechanisms: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR)[19]. When the missingness depend neither on observed data nor on the missing data, the data are said to be MCAR. In this situation, complete cases analysis would yield larger standard errors due to the reduced sample size but does not cause bias as the incomplete dataset is representative of the entire dataset[20]. More often the mechanism of missingness may depend on the observed data, and then the missing data are MAR given the observed data[20]. MAR allows prediction of the missing values based on the cases with complete data. In the situation where the missingness depends on the missing data, then the data are MNAR[21]. Sensitivity analysis[22,23] is an approach recommended by the National Research Council[21] to handle MNAR by assessing the impact of assumptions about the missing data on inference. In this paper, we focus on the mechanism of MAR, which is widely accepted in the biostatistical and health research, and imputation, a statistical technique for handing missing data that has gained wide popularity for its generality. Under MAR, statistical techniques for treating missing data such as the inverse probability weighting[24] or the expectation-maximization algorithm[25] have been used in combination of variable selection. However, these estimating

equations-based or likelihood-based approaches are usually customized to a specific variable selection procedure and are less amenable to general missing data patterns[26]. Wood et al.[27] compared strategies for combining variable selection procedures and imputation for incomplete data. They focused on the backward stepwise selection approach based on the parameter estimates and their standard errors adjusted for imputation. Long and Johnson[26] proposed a general resampling approach that combines bootstrap imputation and stability selection[28], which is based on randomized lasso, and used simulations to demonstrate that this approach had better performance compared with several existing methods for both low- and high-dimensional problems. Both studies rely on parametric modeling of the covariate-outcome relationship.

Some tree-based methods have embedded algorithms for handling missing data. For example, BART and XGBoost can accommodate missing data in covariates by considering the missingness as a value in its own right in the tree splitting rules during model construction[9,29]. This technique is known as missingness incorporated in attributes (MIA). MIA does not require any assumptions or need for imputation, but cannot simultaneously address missing data in the outcome. Some RF based algorithms do not distinguish between outcome and covariates and can deal with missing values in both cases and at the same time. Examples of such algorithms[30] include 1) "strawman imputation", which imputes the missing values for continuous variables using the median of non-missing values and for discrete variables using the most frequently occurring non-missing values, 2) "on-the-fly-imputation", which simultaneously imputes data by randomly drawing a value from in-bag data while growing the forest and iterating for improved results, and 3) imputation algorithm called missForest (more detail in Section 2.3.2), which recasts the missing data problem as a prediction problem. Using simulations, Tang et al.[30] recommended the imputation method missForest, which performed the best across various missing data settings. To the best of our knowledge, no previous research has investigated strategies for combining statistical techniques for missing data and variable selection procedures based on flexible tree-based machine learning methods.

In this paper, we investigate approaches to combining imputation and variable selection procedures for tree-based methods. We consider and compare four machine learning methods that have a proven track record: BART, RF, conditional random forests (CRF)[31] (a variant of RF utilizing conditional inference trees as base learners), and XGBoost, and two commonly used classical parametric regression based methods: the backward stepwise selection approach and lasso. Drawing strength from previous work by Long and Johnson[26], we use bootstrap imputation[32]. It has been shown, when used with lasso, to be relatively insensitive to tuning parameter values in terms of variable selection, as well as the threshold value π that determines which variables should be selected, e.g., if they are selected in at least πM imputed datasets. Motivated by our case study, we focus on a binary outcome. However, all methods considered can be straightforwardly applied to a continuous outcome.

The rest of the paper is organized as follows. Section 2 describes the methods. In Section 3, we conduct an expansive simulation to assess the practical operating characteristics of the methods. Section 4 illustrates the methods using data from the Study of Women's Health Across the Nation, and Section 5 provides a discussion.

2   Methods

2.1   Notation

Suppose we have n independently and identically distributed copies of data $\{(Y_i, \boldsymbol{X}_i), i = 1, \ldots, n\}$, where Y_i is subject i's binary outcome with 1 for case and 0 for control, $\boldsymbol{X}_i = (X_{i1}, \ldots, X_{iK})$ is a set of K potential predictors, either continuous or discrete. Both Y and $\boldsymbol{X}$ can have missing values. Let R_k^x be the missing data indicator for X_k, with $R_k^x = 1$ if X_k is missing and $R_k^x = 0$ if X_k is observed, $k = 1, \ldots, K$, and $\boldsymbol{R}^{\boldsymbol{X}} = (R_1^x, \ldots, R_K^x)$. Similarly, let R^Y be the missing data indicator for the outcome Y, with $R^Y = 1$ if Y is missing and $R^Y = 0$ if Y is observed. Let $\boldsymbol{Q}$ denote the nonlinear transformations and higher-order terms of the predictors $\boldsymbol{X}$, and $\boldsymbol{R}^{\boldsymbol{Q}}$ be the corresponding missing data indicator. We use $\boldsymbol{Z} = (\boldsymbol{X}, \boldsymbol{Q}, Y)$ to denote the full data, $\boldsymbol{Z}^{\text{obs}}$ to indicate the observed components of $\boldsymbol{Z}$ and $\boldsymbol{Z}^{\text{mis}}$ the missing components. The missing data indicator for the full data is $\boldsymbol{R} = \left(\boldsymbol{R}^{\boldsymbol{X}}, \boldsymbol{R}^{\boldsymbol{Q}}, R^Y\right)$. Throughout, we assume that data are MAR, i.e., $f\left(\boldsymbol{R} \mid \boldsymbol{Z}\right) = f\left(\boldsymbol{R} \mid \boldsymbol{Z}^{\text{obs}}\right)$.

2.2   Overview of variable selection methods in complete data

Variable selection has been widely studied from both the classical and Bayesian perspective. A large body of work rests on the assumption taking the shape of parametric models, in which the outcome is explicitly specified in a linear functional form of the covariates. If the parametric form is incorrectly specified, the variable selection results may be erroneous. For example, noise variables may be selected and important variables excluded. In practice, correctly specifying a parametric model in the presence of complex nonlinear functional relationships between the predictors and response is difficult. However, nonparametric methods are flexible to approximate functional forms of arbitrary complexity, thus mitigate the parametric assumptions and attendant errors in variable selection. In this article, we focus on tree-based methods, examples of which include BART, RF, CRF and XGBoost. The tree-based methods use the internals of the classification and regression tree (CART) structure. A CART model starts with a single root node containing all observed data. A splitting rule, involving a splitting variable X_k and a split point c, is chosen for the root node. The data in the root node are then divided into two groups – daughter nodes – based on whether $X_k \geq c$ or $X_k < c$. This procedure is then sequentially performed on each of the daughter nodes until some stopping criteria have been met, leading down to the terminal nodes. See Breiman et al.[10] for a detailed description of the CART. An ensemble of CART models can overcome the instability of single trees and improve prediction accuracy.

2.2.1   Bayesian Additive Regression Trees—BART is a Bayesian sum-of-trees model with a regularizing prior to keep the individual tree effects small[7]. For a binary outcome, BART uses probit regression,

$$\Pr(Y = 1 \mid X = x) = \Phi(f(x)) = \Phi\left(\sum_{j=1}^{m} g(x; \mathcal{T}_j, \mathcal{M}_j)\right),$$

(1)

where $\Phi(\cdot)$ is the standard normal cumulative distribution function, $\mathcal{T}_j$ denotes the jth regression tree with v terminal nodes, each of which has a parameter μ_{jv} representing the mean response of the observations in the dth terminal node ($d = 1, \ldots, v$), and $\mathcal{M}_j = (\mu_{j1}, \ldots, \mu_{jv})$; and $g(\boldsymbol{x}; \mathcal{T}_j, \mathcal{M}_j)$ is the value obtained by dropping the vector of covariates $\boldsymbol{x}$ down the tree $\mathcal{T}_j$ and reporting the mean response μ associated with the terminal node in which $\boldsymbol{x}$ falls. The $(\mathcal{T}_j, \mathcal{M}_j)$ are treated as parameters in a formal statistical model. To avoid overfitting and limit the contribution of each $(\mathcal{T}_j, \mathcal{M}_j)$, a regularizing prior is put on the parameters, and the posterior is computed using Markov chain Monte Carlo (MCMC). At each iteration of the MCMC algorithm, each tree may grow or shrink in size, with its parameters $(\mathcal{T}_j, \mathcal{M}_j)$ possibly swapped with another pair without changing the f. This lack of identification leads to a stable MCMC algorithm. A detailed overview of the BART model can be found in Chipman et al.[7] and Bleich et al.[1].

Variable selection for BART draws on a permutation-based inferential approach[1]. The BART model outputs the "variable inclusion proportions" of each predictor variable: the proportion of times each predictor is chosen as a splitting rule divided by the total number of splitting rules appearing in the model. The variable inclusion proportions can be used to rank variables in terms of relative importance, but do not provide guidelines for variable selection. Bleich et al.[1] proposed a nonparametric approach to establish thresholds for the variable inclusion proportions for a variable to be deemed as important. This approach is based on permutation of the response variable and averages out the chance capitalization that may occur in a single data set. Specifically, P permutations of the response vector are created, $Y_1^*, \ldots, Y_P^*$. The BART model will then be fitted to each of the permuted response vectors Y_p^* and the original predictor variables $(X_1, \ldots, X_K)$. From the BART run using each permuted response Y_p^*, we retain the variable inclusion proportions, $p_{k,p}^*$ for each predictor X_k. Denote the vector of all variable inclusion proportions from the pth permuted response by $\mathbf{p}_p^*$, then the "null" distribution of the variable inclusion proportions across all P permutations can be computed, $\mathbf{p}_1^*, \ldots, \mathbf{p}_P^*$, for the variable inclusion proportions $\mathbf{p}$ from the real response Y. Three thresholding procedures of varying stringency are proposed to select important variables. The least stringent strategy is a "local" threshold: select predictor X_k if p_k exceeds the $1 - \alpha$ quantile of the permutation null distribution of $p_k, p_{k,1}^*, \ldots, p_{k,P}^*$. The most stringent strategy is a "global max" threshold: first calculate $p_{\max, p}^* = \max\{p_{1,p}^*, \ldots, p_{K,p}^*\}$, the largest variable inclusion proportion across all predictor variables in permutation p, and then only select X_k if p_k exceeds the $1 - \alpha$ quantile of the distribution of $p_{\max, 1}^*, \ldots, p_{\max, P}^*$. The intermediary strategy is a "global SE" threshold: calculate the mean m_k and standard deviation s_k of variable inclusion proportion p_k^* for predictor X_k across all permutations, and find the smallest global multiplier C^* such that $C^* = \inf_{C \in \mathbb{R}^+} \left\{ \forall k, \frac{1}{P} \sum_{p=1}^{P} \mathbb{1}(p_{k,p}^* \le m_k + C s_k) > 1 - \alpha \right\}$. The predictor X_k is only selected if $p_k > m_k + C^* s_k$. In our simulations (Section 3), we used the local threshold as the other two thresholds were found to be too stringent for our study settings. We refer to Bleich et al.[1] for more detail on the permutation-based variable selection approach.

2.2.2 Random Forests—The RF method generates a distribution of trees, each constructed on a bootstrap sample. This process is known as *bagging*, short for bootstrap aggregation, first developed by Breiman as one of the earliest ensemble techniques[33]. To reduce correlation among bootstrapped trees and improve the prediction accuracy of the ensemble model, Breiman[8] unified an algorithm called RF, which considers a random subset of predictors for each split in the tree-building process on each bootstrap sample. Certain observations of each bootstrap sample are left out and not used for fitting the tree model. These samples are called out-of-bag (OOB) samples and can be used to evaluate the predictive performance of the RF model.

RF supplies the OOB variable importance score, describing the relative impact of each predictor variable on the model's predictive accuracy. The score of each predictor variable is calculated by permuting that variable's values in the OOB sample while keeping all other variables the same, and recording the difference in prediction accuracy between the permuted and original data. Many variable selection procedures for RF are based on the combination of variable importance and model selection, in particular in the family of the "wrapper" methods. In this article, we use an approach that is based on recursive elimination of variables and has been widely used in the biomedical research.

Following Díaz-Uriarte and De Andres[18], we iteratively fit a series of RF models. In each iteration, a new forest is built after a fraction of predictor variables are discarded with the smallest variable importance scores. The variable importance scores are not recalculated in each iteration to avoid potential overfitting. The OOB error rates from all the fitted RF are recorded. The selected set of variables is the smallest number of variables whose OOB error rate is within u standard errors of the minimum error rate of all RF models (the standard error is computed from the exact binomial distribution with the rate parameter being the minimum error rate). Setting $u = 0$ corresponds to selecting the set of variables that leads to the smallest error rate; and setting $u = 1$ is similar to the "1 s.e. rule" commonly used in the classification literature. We use $u = 1$ in our simulations (Section 3) and case study (Section 4).

2.2.3 Conditional Random Forest—CRF is a variant of the RF model with the conditional inference trees of Hothorn et al.[31] as base learners. In a conditional inference framework, statistical hypothesis tests are utilized to conduct an exhaustive search across the predictor variables and the possible split points of these variables. For a candidate split, a statistical test with a *p*-value computed is used to evaluate the between-group difference for the two nodes generated by the split. A threshold for statistical significance is used as the stopping criterion to determine whether additional splits should be performed. Like CART trees, conditional inference trees can be bagged. In constructing the ensembles, observations are sampled according to probabilities specified by the weights. The CRF also differs from RF in the aggregation scheme. The CRF algorithm averages observation weights extracted from each of the conditional inference trees rather than direct predictions as RF. For variable selection using CRF, we also employ the recursive backward elimination approach as used for RF, based on the variable importance score and OOB error rate outputted by CRF. Hothorn et al.[31] and Strobl et al.[34] argued that computation of variable importance is biased

in favor of variables with many potential cutpoints in RF; on the contrary, the conditional inference trees used in CRF are unbiased.

2.2.4 Extreme Gradient Boosting

The XGBoost method draws on the idea of gradient boosting[9]. Boosting is a process in which a weak learner is boosted into a strong learner. Friedman et al.[35] connected boosting to a forward stagewise additive model that minimizes a loss function and brought forth a highly adaptable algorithm, gradient boosting machines. Specifically, a gradient tree boosting model uses m additive functions to predict the outcome,

$$\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) = \phi(\boldsymbol{x}) = \sum_{j=1}^{m} f_j(\boldsymbol{x}),$$

(2)

where $f = w_{q(x)}$ is a tree structure with decision rules q that map an observation $\boldsymbol{x}$ to a terminal node, and w represents the weights associated with terminal nodes. Each tree contains a continuous score on each of the terminal nodes. For a given $\boldsymbol{x}$, we will drop $\boldsymbol{x}$ down each of the m boosted trees until it hits a terminal node of each tree, and calculate the final prediction for $\boldsymbol{x}$ by summing up the scores (given by w) in the corresponding terminal nodes across the m trees, i.e. $\sum_{j=1}^{m} f_j(\boldsymbol{x})$ in equation (2). The idea of gradient boosting is to minimize the loss function $\mathscr{L}(\phi) = \sum_i l(y_i, \hat{y}_i)$, where $\hat{y}_i$ and y_i are respectively the predicted and observed outcome for the ith individual in the data. For classification, $l(y_i, \hat{y}_i) = \exp(-y_i \hat{y}_i)$ is an exponential loss function (differentiable convex) that measures the difference between $\hat{y}_i$ and y_i, $\forall i \in \{1, \ldots, n\}$. The XGBoost algorithm adds a term $\Omega(f_j)$ to the loss function $\mathscr{L}(\phi)$ of the traditional gradient tree boosting to penalize model complexity. The revised loss function for XGBoost is

$$\mathscr{L}(\phi) = \sum_{i=1}^{n} l(\hat{y}_i, y_i) + \sum_{j=1}^{m} \Omega(f_j),$$

(3)

where $\Omega(f) = \gamma T + \frac{1}{2}\lambda \parallel w \parallel^2$, T is the number of terminal nodes in a tree, $\parallel w \parallel$ represents the ℓ_2 norm of terminal node weights, γ and λ are tuning parameters governing further partitioning of the predictor space. The penalty term $\Omega(f)$ helps to smooth the final learnt weights w to avoid overfitting. XGBoost uses two additional techniques, shrinkage and column subsampling, to further prevent overfitting. Shrinkage scales newly added weights by a factor after each step of tree boosting, as in traditional gradient tree boosting. The column subsampling technique is borrowed from the RF algorithm, which selects a random subset of predictors for split in each step of tree boosting[9].

To the best of our knowledge, there is no principled method to variable selection using XGBoost. We propose to leverage the variable importance score provided by XGBoost and apply the recursive feature elimination procedure used for RF to select important

predictors with XGBoost. Since XGBoost does not have OOB samples, in our simulation study (Section 3), we evaluate the model classification error on a 50% hold-out set for $n = 1000$ and using 5-fold cross-validation for $n = 250$.

2.2.5   Least absolute shrinkage and selection operator—Shrinkage methods are frequently used for variable selection by regularizing the coefficient estimates. The lasso is among the most popular shrinkage methods[36]. Variable selection via lasso for a binary outcome is by maximizing a penalized version of the log-likelihood of a logistic regression,

$$\max_{\beta_0, \boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} \left[y_i \left(\beta_0 + \boldsymbol{\beta}^\top \boldsymbol{x}_i \right) - \log \left(1 + e^{\beta_0 + \boldsymbol{\beta}^\top \boldsymbol{x}_i} \right) \right] - \lambda \sum_{k=1}^{K} |\beta_k| \right\},$$

(4)

where β_0 is the intercept term of the logistic regression model, and $\boldsymbol{\beta}$ corresponds to a vector of coefficients of $\boldsymbol{X}$. Criterion (4) is concave, and a solution of $\{\beta_0, \boldsymbol{\beta}\}$ can be found using nonlinear programming methods. The ℓ_1 penalty in (4) has an effect of forcing some β_k's to be exactly equal to zero when the tuning parameter λ is sufficiently large. In this sense, the lasso selects the best subset of predictors and hence performs variable selection.

2.2.6   Backward Stepwise Selection—Classical variable selection is usually implemented through recursive procedures such as forward, backward, and stepwise selection. Both backward and forward selection methods have drawbacks, and stepwise selection provides a compromise. The backward stepwise selection starts with a model with all potential predictors. The backward selection process removes from the model the predictor that has the least impact on the fit (e.g., at α significance level). The forward selection process checks whether removed variables should be added back into the model (e.g., at α_2 significance level, $\alpha_2 = \alpha(1 - \epsilon)$ for ϵ small). There are other stepwise selection procedures that use Z-score or the Akaike information criterion for the inclusion and deletion processes[37]. In this paper, we use the same procedures described in Wood et al.[27], which are based on statistical testing.

2.3   Variable selection with missing data

2.3.1   Generating missing data—To accurately mimic realistic missingness problems, we use the multivariate amputation approach described in Schouten et al.[38] It has been shown that stepwise univariate amputation procedure – generating missing values for one variable at a time – may not appropriately control the missingness percentage and may lead to an inflated observed data skewness when the missing data proportion increases[38]. The multivariate amputation procedure generates any missing data scenario precisely as desired with respect to the missingness percentage and the missing data mechanism. In addition, the degree of skewness in the observed data is not sensitive to the number of amputated variables. These operating characteristics allow for valid and fair evaluations of variable selection methods in the presence of missing data.

The multivariate amputation procedure first randomly divides the complete data into a certain number (can be one) of subsets based on the assumed number of missing data patterns. The size of these subsets may vary. The use of subsets allows the specification of any missing data pattern for any subgroups while establishing the desired missingness percentage in the merged data. For example, in our case study (Section 4), participants with certain characteristics may be reluctant to answer questions related to their dietary habits, which will lead to joint missingness in all dietary variables (e.g., dietary intakes of isoflavones, lignans or coumestrol)[39]. This missingness characteristic can be emulated by generating joint missingness in several variables in a subset of data. The weighted sum scores are then calculated for individuals in each subset to amputate the data. Specifically, the weighted sum score for an MAR predictor X_k and individual i of a subset follows

$$wss_{x_k, i} = X_i w_1 + Q_i w_2 + Y_i w_3, w_1, w_2, w_3 \in \mathbb{R}$$

(5)

where w_1, w_2 and w_3 are respectively a vector of user-specified weights for the untransformed versions of the predictors X_i, transformed versions of the predictors captured in Q_i and the outcome Y_i. Note that the weights are the same for individuals in the same subset (assumed to have the same missing data pattern). To induce MAR, a zero weight is assigned to the predictor that will be amputated. For example, if $R_k^X = 1$, then the weights for X_k and its nonlinear transformation and higher order terms will be set to zero; while nonzero weights will be assigned to the other predictors and their nonlinear counterparts $X_{-k}^s \subseteq (X_{-k}, Q_{-k})$, on which $p(R_k^X \mid X_{-k}^s)$ depends. A logistic distribution function, $\Pr(R_k^X = 1) = \text{logit}^{-1}(g(X_{-k}^s))$ is then applied on wss_i to compute the missingness probability, which is used to determine whether the data point becomes missing or not. Different $g(\cdot)$ functions can be used to create different missing data patterns. For example, the standard logistic function $g(x) = 1/(1 + e^{-x})$ generates a right-tailed type of missingness, that is, individuals with high weighted sum scores will receive a high probability of being missing; $g(x) = -a + |x - b| \forall a \in \mathbb{R}^+, b \in \mathbb{R}$ creates a both-tailed missingness type giving higher missingness probabilities to individuals with extreme weighted sum scores[40]. Putting all the subsets together, we will have a complete data set with desired missingness characteristics.

2.3.2 Variable selection in bootstrap imputed data—Following Long and Johnson[26], we first conduct bootstrap imputation – originally developed by Efron[32] – before variable selection. Wood et al.[27] investigated methods for combining multiple imputation and the backward stepwise variable selection. Their recommended approaches relied on combined inference about parameter estimates to consolidate variable selection results. These approaches, however, are not amenable to machine learning modeling techniques as the parametric functional forms are not required in machine learning models. Another popular approach is to perform variable selection on each of the M imputed datasets, and then select predictors if they appear in at least πM models. Results in Wood et al.[27] show that this approach, when combined with multiple imputation, can be sensitive to the threshold value π. The numerical studies in Long and Johnson[26] suggest that when

combined with bootstrap imputation and stability selection, this approach can be relatively insensitive to π and tuning parameter values used in variable selection. Stability selection is a general variable selection approach based on resampling or subsampling for fully observed data[28]. In stability selection, the randomized lasso is applied to each random sample (by random subsampling or resampling) of the observed data, and important predictors are selected based on the variable selection results from all random samples using a threshold π. We investigate the operating characteristics of bootstrap imputation used in conjunction with machine learning based variable selection procedures. The bootstrap imputation procedure is described as follows:

1. Generate B bootstrap datasets $\{ \mathbf{Z}^{(b)}, b = 1, ..., B\}$, each of which has a missing indicator $\mathbf{R}^{(b)}$ corresponding to the observed data $Z^{\mathrm{obs}(b)}$.

2. Conduct imputation for each bootstrap data set $\{\mathbf{Z}^{(b)}, \mathbf{R}^{(b)}\}$ using an imputation method of choice. The imputed and complete datasets are denoted by $\{\mathbf{Z}^{\mathrm{imp}(b)} = (Y^{(b)}, \mathbf{X}^{(b)}), b = 1, ..., B\}$.

A single imputation is performed for each bootstrap sample. We use two methods for imputation in Step 2: (i) the standard imputation program using the R package `mice`[41], and (ii) RF based imputation algorithm using the R package `missForest`[42]. Imputation via `mice` uses the iterated chained equations approach[43], which requires specifying conditional models for each incomplete variable given all other variables and drawing imputations by iterating over the conditional distributions. The `missForest` approach imputes data by growing a forest and regressing each variable that has missing values in turn against all other variables and then predicting missing data for the dependent variable using the fitted forest[30]. The `missForest` method has been shown to have a better imputation performance than the parametric `mice` approach[44]. We investigate whether imputation via nonparametric `missForest` can improve variable selection with missing data over parametric `mice`. Variable selection via each of the methods considered will be performed for all $b = 1, ..., B$ bootstrap imputed datasets. Denote by $\hat{\mathcal{S}}^{(b)}$ the selected set of predictors on the bth bootstrap imputed data. Then the final selected set of predictors $\hat{\mathcal{S}}_{\pi} \subseteq \mathbf{X}$ is

$$\hat{\mathcal{S}}_{\pi} = \{ X_k : \Pi_k \geq \pi \},$$

(6)

where $\Pi_k = (1/B) \sum_{b=1}^{B} \mathbb{1} \left(X_k \in \hat{\mathcal{S}}^{(b)} \right)$ and $\pi \in (0, 1)$ is a fraction threshold for selecting a predictor. Higher values of π correspond to more stringent rules for selecting the important variables, whereas lower values of π can lead to larger sets of selected variables, which may include noise predictors. In our simulations (Section 3), We explore the performance of all methods considered using the threshold values from the fine resolution grid on the interval [0, 1].

2.4 Performance metrics

Each of the methods considered will be compared on the ability to select predictor variables that are truly associated with the outcome, or useful predictor variables. We use the following five performance metrics commonly used in the variable selection literature[1,27].

1. Precision $= \frac{TP}{TP + FP}$, where TP and FP are respectively the number of true positive and false positive selections. The *precision* of a variable selection method is the proportion of truly useful predictors among all selected predictors.

2. Recall $= \frac{TP}{TP + FN}$, where FN is the number of false negative selections. The *recall* of a variable selection method is the proportion of truly useful variables selected among all useful variables. This is sometimes referred to as the *power* in the literature[27].

3. $F_1 = \frac{2\,\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$. The F_1 measure is the harmonic mean of precision and recall, balancing a method's ability to avoid selecting irrelevant predictors (precision) with its ability to identify the full set of useful predictors (recall).

4. Type I error. A variable selection method's *Type I error* is defined as the mean of the probabilities that the method will incorrectly select each of the noise predictors.

A good method should have high values of precision, recall and F_1, and low Type I error (which may differ from the nominal significance level used in some model selection procedures)[27].

2.5 Software implementation

The multivariate amputation procedure was conducted using the `ampute` function in the R package `mice`[38]. Parametric imputation was performed using the R package `mice`[41]. We used the default methods: predictive mean matching for numeric variables and logistic regression imputation for binary variables. Iterative series of predictive models were used. In each iteration, each specified variable in the dataset is imputed using the other variables in the dataset. The number of iterations was set as five and the number of imputations was set to be one (since we use bootstrap imputation). The nonparametric imputation was performed using the R package `missForest`[42]. The default settings were used: the maximum number of iterations to be performed = 10, the number of trees = 100, and `mtry` (the number of randomly selected variables for splitting) = $\sqrt{K}$. We used the R package `bartMachine` to perform variable selection for BART. For the BART models, we used 1100 draws with the first 100 discarded as burn-in. The permutation distribution was obtained from 100 BART model constructions. To implement variable selection using RF, the R package `varSelRF` was used[18]. We used the `cforest` function in the R package `partykit`[11] to fit the CRF models. To build RF and CRF models for recursive elimination of predictors, we used 5000 trees for the first full model and 2000 trees in each iteration step. The fraction of variables from those in the previous forest to exclude at each iteration was set to `vars.drop.frac = 0.1`, `mtry` (the number of randomly selected variables) was set at $\sqrt{K}$ and default

values for other tuning parameters (e.g., `nodesize =1` and `maxnodes` was not specified to allow trees to be grown to the maximum possible for RF; `minbucket = 7`, `minsplit = 20` for CRF) were used. The XGBoost models were fitted using the `xgboost` function in the R package `xgboost`[45]; the max number of tree boosting iterations was set to `nrounds = 200` and the fraction of variables to be excluded in each iteration was set as 0.1. We implemented the lasso using the R package `glmnet`[46]. The tuning parameter `lambda` was selected as the largest value such that the mean cross-validated deviance was within one standard error of the minimum deviance. R codes to implement stepwise selection and our proposed variable selection algorithms using XGBoost and CRF, and to replicate our simulation studies are provided in the GitHub page of the first author https://github.com/liangyuanhu/Variable-selection-w-missing-data.

3   Simulation

3.1   Simulation design

The simulation scenarios are motivated by our case study to represent data structures commonly observed in health studies. We considered two sample sizes, small sample size $n = 250$ and large sample size $n = 1000$. We generated 10 useful predictors that are truly related to the responses, $X_1, ..., X_{10}$, and 40 noise predictors, $X_{11}, ..., X_{50}$. This ratio of useful versus noise predictors was chosen to mimic our case study. In Section 3.2, we vary the number of irrelevant predictors from 10 to 100 and examined how different ratios will impact the performance of the methods considered. We drew independently X_1 and X_2 from Bern(0.5), X_3, X_4 and X_5 from the standard $N(0, 1)$, X_6 from Gamma(4, 6), and X_7, X_8, X_9, X_{10} were designed to have missing values under the MAR mechanism. For an MAR predictor, we specify the true forms in which the predictor depends on the other predictors. Both nonlinearity and nonadditivity were considered for the dependence structure among the predictor variables to compare the imputation methods `missForest` and `mice`. Specifically, the following are the true data models for X_7, X_8, X_9, X_{10}:

$$x_7 \mid x_5, x_6 \sim N(-0.4x_5 + 0.4x_6 + 0.3x_5x_6, 1)$$
$$x_8 \mid x_5, x_6, x_7 \sim N(0.1x_5(x_6 - 2)^2 - 0.1x_7^2, 1)$$
$$x_9 \mid x_3, x_4, x_5 \sim N(0.5x_3 + 0.3x_4 - 0.3x_5^2 + 0.2x_3x_4, 1)$$
$$x_{10} \mid x_3, x_4, x_5, x_9 \sim N(0.1x_3^3 - 0.3x_4 - 0.4x_5 + 0.2x_9^2 + 0.3x_4x_5, 1).$$

We further generated 20 continuous noise predictors $x_{11}, ..., x_{30} \overset{i.i.d}{\sim} N(0, 1)$, and 20 binary noise predictors $x_{31}, ..., x_{50} \overset{i.i.d}{\sim}$ Bern(0.5). The true outcome model is specified as the following:

$$\Pr(y = 1 \mid x_1, ..., x_{10}) = \text{logit}^{-1}\Big(-2.7 + 1.8x_1 + 0.5x_2 + 1.1x_3 - 0.4e^{x_5} - 0.4(x_6 - 3.5)^2 + 0.3(x_7 - 1)^3 + 1.1x_8$$
$$-1.1x_{10} + 5\sin(0.1\pi x_4 x_9) - 0.4x_5x_{10}^2 + 0.4x_3^2x_8 \Big).$$

To avoid any concerns that we are deliberately selecting simulation settings where machine learning methods will show good performance, we considered the outcome model with

arbitrary data complexity that reflects common situations in health datasets: (i) discrete predictors with strong (x_1) and moderate (x_2) associations; (ii) both linear and nonlinear forms of continuous predictors; (iii) nonadditive effects ($x_4 x_9$).

After generating the full data, we amputated X_7, X_8, X_9, X_{10} and Y under MAR using the multivariate amputation approach described in Section 2.3.1. Previous studies[47,48] show that multiple imputation could provide unbiased results when the proportion of missing data is up to 90%. We investigate the performance of bootstrap imputation based variable selection methods across three levels of missingness: (i) 40% missingness in Y and 60% overall missingness; (ii) 15% missingness in Y and 30% overall missingness; and (iii) 7.5% missingness in Y and 15% overall missingness. Scenario (i) mimics the percentages of missingness observed in our case study (Section 4). Note that each variable can still have a small-to-moderate proportion of missing values. The percentage of the cases that are made incomplete indicates the overall missingness. For scenario (i), we first randomly divided the full data into 8 subsamples, which were the following percentages of the whole data: 0.30, 0.09, 0.09, 0.08, 0.08, 0.16, 0.10, and 0.10. Each subsample was then amputated to have a specific missing pattern using the weighted sum scores defined in equation (5). The weighted sum scores relate the missingness on amputated variables to the values of other variables as follows:

1. $wss_{y,i} = 5x_1 + 5x_2 + x_3 - x_5 - x_6 + x_7 + x_8 + x_{10} - 0.5x_6^2 + 1.5x_4 x_9 - 0.5x_5 x_{10} + 0.5x_3 x_8$

2. $wss_{x7,i} = x_5 + x_6 + x_5 x_6$

3. $wss_{x8,i} = 5y + x_5 + x_6 + x_7 + x_7^2 + x_5 x_6$

4. $wss_{x9,i} = 5y + x_3 + x_4 + x_5 + x_5^2 + x_3 x_4$

5. $wss_{x10,i} = 5y + x_3 + x_4 + x_5 + x_9 + x_4 x_5$

6. $wss_{y,x7,x8,i} = x_5 + x_6$

7. $wss_{y,x8,x10,i} = x_5$

8. $wss_{y,x9,x10,i} = x_3 + x_4 + 0.5x_5^2 + 0.5x_3 x_4$

The weighted sum score gives a nonzero weight to the variables and their nonlinear forms and interactions therein, on which the probabilities to be missing for amputated variables depend. In subsample (1), the responses were amputated. The predictor variables X_7, X_8, X_9, X_{10} were respectively amputated in subsample (2), subsample (3), subsample (4), and subsample (5). We further created the joint missingness in (Y, X_7, X_8) in subsample (6), (Y, X_8, X_{10}) in subsample (7), and (Y, X_9, X_{10}) in subsample (8). Finally, we applied the logistic distribution function to the weighted sum scores to create the missing indicators and amputate data. A right-tailed type of missingness was used for subsamples (1)–(5) and a both-tailed type of missingness was used for subsamples (6)–(8). In each subsample, there were 60% missing values in the amputated variables. The amputated eight subsamples were combined to form a whole dataset. This setup creates 40% missingness in Y and 60% overall missingness. The scenario (iii) with 7.5% missingness in Y and 15% overall missingness was generated by creating 15% missing values in the amputated variables. We

generated missingness scenario (ii) with 15% missingness in Y and 30% overall missingness by changing the proportions of the eight subsamples to 0.20, 0.15, 0.15, 0.10, 0.10, 0.10, 0.10, and 0.10 and creating 30% missing values in each subsample.

There are a total of 8 scenarios considered in our simulation: 2 sample sizes ($n = 250$ and $n = 1000$) × 4 percentages of missingness (complete data, 15% overall missingness, 30% overall missingness and 60% overall missingness). For computational considerations, we replicated each simulation independently $M = 250$ times for $n = 1000$ and $M = 1000$ times for $n = 250$. For each Monte Carlo dataset, we draw $B = 100$ bootstrap samples and perform imputation on each sample. In Section 3.2, we additionally investigated the sensitivity of complete data performance – a benchmark against which performance on incomplete data will be compared – to the number of noise predictor variables.

3.2   Simulation results

When implementing the methods, we included all 50 predictor variables (10 useful and 40 noise predictors) and the outcome variable available to the analyst in the imputation models for both `mice` and `missForest`. The top panel of Table 1 displays four performance metrics for each of six methods considered on complete data with $n = 250$ and $n = 1000$. For the large sample size $n = 1000$, four machine learning methods, BART, XGBoost, CRF and RF outperformed the two parametric methods, lasso and backward stepwise selection. BART, XGBoost, CRF and RF all had good performance, with BART being the top performer and CRF and RF having relatively lower recall. Between the two parametric methods, lasso tended to have higher precision and backward stepwise selection tended to higher recall and higher type I error. A perusal of the recall for each of 10 useful predictor variables (Web Figure 1) shows that lasso and backward stepwise selection failed to identify the two nonadditive variables X_4 and X_9, and had a low probability of selecting X_{10}, which had complex forms of relationship with other predictors and with the response. On the other hand, BART and XGBoost had a high success rate of identifying all continuous variables in conditions of nonadditivity and nonlinearity, followed by CRF and RF, but all four machine learning based methods were less likely to select discrete variables compared to parametric methods. All methods had deteriorated performance, with no apparent "winning" method when the sample size decreased from $n = 1000$ to $n = 250$, demonstrated by lower precision, recall and F_1 and higher Type I error. The recall or power, in particular, had a substantial drop, suggesting the difficulty in recovering the full set of useful predictors with a small sample size.

With MAR covariates and outcome, our proposed variable selection procedures can recover the performance achieved on complete data. Figure 1–4 respectively compare the curves of precision, recall, F_1 and type I error over the interval $\pi = (0, 0.1, \ldots, 1)$ for the six methods considered and for $n = 1000$. The variable selection results on incomplete data vary with the threshold value of π, with the precision increasing and the recall and type I error decreasing over the interval of π. Overall, BART, CRF and RF have the highest precision and the lowest type I error; backward stepwise selection and lasso have the highest recall but also the highest type I error. Figure 3 presents the F_1 score, a balance of precision and recall, for each of three missingness proportions and each of two imputation methods. The optimal

value of π yielding the highest F_1 varies by methods, but remains largely consistent across a combination of scenarios of missingness proportions and imputation methods. BART, CRF and RF achieved the best F_1 at $\pi = 0.1$ or $\pi = 0.2$, XGBoost attained the highest F_1 with $\pi = 0.4$ or $\pi = 0.5$, while lasso and backward stepwise selection favored a larger threshold value around $\pi = 0.8$. The performance of all methods deteriorated when the proportions of missing data increased. The nonparametric imputation technique, `missForest` did not improve the performance compared to `mice`, except for BART with 60% missingness. Figure 5 shows the power of each method for selecting each of the 10 predictors in the Y model. XGBoost has the lowest power among all methods for detecting discrete variables (X_1 and X_2) across all scenarios, but has superior performance in identifying complex nonadditive (X_4 and X_9) and nonlinear (X_{10}) continuous variables. BART appears to have consistently good performance across all data types, except that in the scenario of 60% missing data imputed via `mice`, it has a poor power in detecting X_4 and X_9. While having the best performance in selecting discrete variables, lasso and backward stepwise selection still have a low power when the effect size of a discrete variable is moderate (X_2). When the sample size reduced to $n = 250$, the performance of all methods deteriorated, but XGBoost appeared to have the smallest drop in the power for detecting the useful predictors, as demonstrated in Web Figure 2.

Web Figure 3–6 respectively display precision, recall, F_1 and type I error for $n = 250$. The overall patterns of the four performance metrics remain similar to those for $n = 1000$. Although our variable selection procedures on incomplete data can still recover the performance achieved on complete data, with substantially reduced sample size, none of the methods could produce satisfactory performance even with moderate proportion of missing data. Machine learning methods had a larger drop in performance compared to parametric methods. Among the four tree-based machine learning methods, XGBoost appears to be the least impacted and still outperforms the lasso and backward stepwise selection.

Variable selection results on incomplete data with 15%, 30% and 60% overall missingness are summarized in Table 1 for all six methods considered, using the imputation method `mice`. For each method, we show results for threshold values of π leading to the highest F_1 and lowest type I error as well as results on the complete cases. Overall, four tree-based machine learning methods produced better performance than lasso and backward stepwise selection, demonstrated by higher F_1 and lower type I error. Using the performance on complete data as a benchmark, all methods can recover the performance benchmark when the missingness proportion is small (15%) or moderate (30%) for both sample sizes. With increased proportion of missing data, BART had the largest drop in performance even with a large sample size, and XGBoost appears to be the only method that can achieve the performance benchmark even with a small sample size. Variable selection results using `missForest` as the imputation method are summarized in Web Table 1. The optimal threshold value for π tended to be smaller when using `missForest`, and BART had a better performance when there is a large proportion (60%) of missing data. Overall, there is no evident improvement in performance over using `mice`. For all methods, variable selection via bootstrap imputation yielded better performance than variable selection amongst the complete cases across all scenarios.

Finally, Figure 6 compares the distributions of the numbers of selected noise and useful predictor variables across 250 replications for each of six methods in each of six scenarios defined by missingness proportion and imputation methods and for $n = 1000$. Compared to the two parametric methods, lasso and backward stepwise selection, four machine learning methods were able to identify more useful predictors across all scenarios, while generally selecting less noise predictors. Among machine learning methods, the number of selected useful predictors is substantially larger in the case of 15% missing data than in the case of 60% missing data, for both imputation methods, mice and missForest. With the same missingness proportion, there is no apparent "winner" between the two imputation methods.

Results of a sensitivity analysis exploring the impact of the number of noise predictors on complete data performance of six variable selection methods are shown in Figure 7. The RF and CRF are the only two methods that are comparatively insensitive to the number of noise predictors. For BART, the recall and F_1 are highly sensitive to the number of noise predictors; the best performance is achieved when the number of noise predictors is > 30 (the ratio of useful versus noise predictors > 1/3), and the performance is substantially worse than other methods when the number of noise predictors reduces to 10. XGBoost consistently yields good performance, which is generally insensitive to the number of noise predictors; the Type I error is slightly higher when there are less than 20 noise predictors. Overall, backward stepwise selection and lasso produce lower precision and F_1, and higher type I error; and for backward stepwise selection, precision decreases at a fast rate as the number of noise predictors increases. Web Figure 7 shows disaggregated performance of each method for $X_1 - X_{10}$. The high sensitivity of BART to lower numbers of noise predictors is manifested in several variables (both discrete and continuous variables). Machine learning methods are less capable of identifying discrete variables, even those with large effect size, but are much more adept at detecting nonlinearity and nonadditivity.

4   Case study: The Study of Women's Health Across the Nation

We analyzed a data set from the Study of Women's Health Across the Nation (SWAN). The SWAN study was a multicenter, longitudinal study aiming to understand women's health across the menopause transition. The SWAN data set contains 3302 women of five racial/ethnic groups aged between 42 and 52, who were enrolled in 1996–1997 from seven sites of the US and were followed to 2018 annually. More detail about the SWAN study can be found in Janssen et al.[49] Despite growing research using the SWAN study data in various areas of women's health, there is a dearth of robust studies identifying key predictors for health outcomes such as metabolic syndrome in the presence of both missing covariates and outcomes.

Metabolic syndrome is a cluster of conditions that occur together, representing "de-tuning" of metabolic adaptations, and has been shown to increase the risk of heart disease, stroke and type 2 diabetes[50]. Prior work[51,52] has shown that the incidence of metabolic syndrome is associated with various risk factors including blood pressure, triglycerides, waist circumstance, triglycerides, glucose, body mass index, waist to hip circumference ratio, and lipoprotein(a). However, these studies were not specifically focused on the health of women during their middle years, which was the target population of our study. In

addition, it is possible that there exists important risk factors that have not been identified previously.

We sought to identify predictors of 3-year incidence of metabolic syndrome. Metabolic syndrome is defined as the presence of at least three of the following five symptoms: abdominal obesity, hypertension, hypertriglyceridemia, impaired fasting glucose, and low high-density lipoprotein cholesterol level[50]. Our analysis included 2313 women who did not have metabolic syndrome at enrollment. Among the 2313 women, 251 (10.9%) developed metabolic syndrome within three years of enrollment, 1240 (53.6%) did not, and the remaining 822 (35.5%) had missing outcomes. Based on previous literature[49,50,53,54] on risk factors for metabolic syndrome, we selected 60 candidate predictors (29 continuous variables and 31 discrete variables), including demographics, daily life behaviour, dietary habits, sleep habits, medications, mental status, menopausal status and related factors, physical measurement, blood measurement, and bone mineral density. A list of 60 variable names and their definitions are displayed in Web Table 2. Only 11 variables are fully observed; the amount of missing data in the other variables ranges from 0.1% to 27.1%. Only 1047 (45.3%) participants have observed data for all predictors and 763 (33.0%) for all predictor and outcome variables.

We conducted imputation on 100 bootstrap samples and implemented six variable selection methods: BART, XGBoost, CRF, RF, backward stepwise selection and lasso. All 60 candidate predictors and the outcome variable were included in the imputation models for both `mice` and `missForest`. The imputation technique and the threshold value of π, which gave the best performance for each of the methods in the simulation study under the scenario representative of the SWAN data structure ($n = 1000$ and 60% overall missingness), were chosen for each respective method. Table 2 summarizes the variable selection results. BART and XGBoost identified the largest set of predictors (17 and 18); while lasso and backward stepwise selection selected the least predictors (10). Four variables, diastolic blood pressure (DIABP), systolic blood pressure (SYSBP), lipoprotein(a) (LPA) and triglycerides (TRIGRES), were selected by all of the six methods. Additional four variables were selected by all four machine learning methods, body mass index (BMI), tissue plasminogen activator (TPA), waist circumference (WAIST) and waist to hip circumference ratio (WHRATIO), among which, TPA and WHRATIO were not identified by the two parametric methods. The backward stepwise selection approach selected more variables that were not chosen by any of four machine learning methods than the lasso. A full list of names and definitions for the 60 candidate predictor variables can be found in Web Table 2.

As an anonymous reviewer pointed out, in the situation where it remains unclear which method is able to select the most relevant predictors, one may use the cross-validated error to help distinguish between the methods. We evaluated the prediction performance, based on area under the curve (AUC), Cohen's Kappa statistic and misclassification error, of the six models that regress the outcome variable on their respectively selected predictor variables. Figure 8 shows the distribution of 5-fold cross-validated AUC among 100 bootstrap samples with imputation performed via the best technique suggested in simulations for each of six methods. BART boasts the highest AUC, closely followed by XGBoost; the lasso and backward stepwise selection have the lowest AUC. Displayed in Web Figure 8–9 are the

distributions of 5-fold cross-validated Cohen's Kappa and misclassification error, which suggest the same performance rankings of the methods. In light of these results, variables selected by BART may be an important addition to the literature. Among the 17 variables selected by BART, some have been less commonly identified as risk factors for metabolic syndrome in the literature, but selection of these variables may be supported by domain knowledge. For example, levels of TPA antigen and low apolipoprotein A-1 (APOARES) were found to be associated with insulin resistance, which was involved in the pathogensis of impaired fasting glucose[54,55].

5 Discussion

We investigate a general variable selection approach when there are missing data in both covariates and outcomes. This approach exploits the flexibility of machine learning modeling techniques and bootstrap imputation, which is amenable to nonparametric methods in which the effect sizes of predictor variables are not naturally defined as in parametric models. Our numeric results show that the proposed variable selection procedure based on three machine learning methods: BART, RF and XGBoost achieves good practical operating characteristics. When the sample size is sufficiently large (e.g., $n = 1000$), even with a large amount of missing data, the variable selection procedure can recover the performance achieved on complete data. The proposed approach can be readily applied to a variable selection problem with a general missing data pattern.

Several considerations must be taken into account in choosing an "optimal" variable selection method. First, an investigator needs to decide the goal of variable selection and choose the corresponding performance metrics: whether it is to identify as many useful predictors as possible (precision), or to avoid selecting irrelevant predictors (recall), or to achieve the best balance between these two goals (F_1 and type I error). Second, while BART has the best performance across various scenarios in our simulation study, our exploratory sensitivity analysis on complete data shows that BART is sensitive to the ratio of useful versus noise predictors. When the ratio is relatively large (e.g., 1:1), BART has a poor recall and F_1. As the ratio decreases to $< 1/3$, BART achieves the best balance between selecting useful predictors and avoiding irrelevant predictors. On the other hand, XGBoost shows good performance across various scenarios and is generally insensitive to the amount of noise information, but has a low power of detecting discrete predictors. RF is highly capable of avoiding selecting unimportant predictors but is less capable of uncovering the full set of important predictors. In practice, it may be useful to run several variable selection approaches and if the chosen variable selection performance metrics are similar across methods, then a cross-validated error estimation of the model with selected variables may help further distinguish between methods, as demonstrated in our case study. Third, while adept at detecting nonlinear and nonadditive variables, machine learning methods have lower power of detecting discrete variables, even those with relatively strong effects, compared to parametric models such as lasso and backward stepwise selection. As a result, machine learning methods may be less favored if a majority of candidate predictors are discrete.

One limitation of our bootstrap imputation-based variable selection procedure is the computational cost of running machine learning models on a large data set. However, it

should be noted that the bootstrap resampling can be computed in parallel on multiple cores when such resources are available. A challenging but important avenue for future research is developing inference-based variable selection methods using the variable importance measure provided by machine learning models. An immediate next step could be to extend the BART-based approach by leveraging the Bayesian posterior samples of the variable inclusion proportions[56]. When using multiple imputation for missing data in conjunction with BART models, we can use Rubin's rule[27,57] or pooled posterior samples arising from the multiple datasets[58] to combine inferences for variable inclusion proportions, which can be further used for variable selection. It is less straightforward for frequentist machine learning methods. Ishwaran and Lu[59] proposed a subsampling approach to estimate the variance and construct confidence intervals of RF's variable importance scores. However, in the presence of missing data, the reduced sample size in a random subsample should have an adverse effect on imputation as it depends on the observed data[26]. One possible strategy is to derive a nonparametric extension of the usual ANOVA-derived measure of variable importance in parametric models. Particularly, Williamson et al.[60] discussed a generalization of the ANOVA variable importance measure, $\Psi_s(P) = \int \{\mu_P(x) - \mu_{P,s}(x)\}^2 dP(x)/var_P(Y)$, where $\mu_P(x)$ and $\mu_{P,s}(x)$ are respectively the conditional mean of outcome Y given the full set and a subset of covariates, under the data-generating mechanism P. Further research in adapting this measure into the settings where missing data are present could be a worthwhile contribution.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Funding

References

1. Bleich J, Kapelner A, George EI et al. Variable selection for BART: an application to gene regulation. The Annals of Applied Statistics 2014; 8(3): 1750–1781.

2. Hu L, Liu B, Ji J et al. Tree-based machine learning to identify and understand major determinants for stroke at the neighborhood level. Journal of the American Heart Association 2020; 9(22): e016745. [PubMed: 33140687]

3. Hapfelmeier A and Ulm K. A new variable selection approach using random forests. Computational Statistics & Data Analysis 2013; 60: 50–69.

4. Hu L, Ji J, Li Y et al. Quantile regression forests to identify determinants of neighborhood stroke prevalence in 500 cities in the USA: implications for neighborhoods with high prevalence. Journal of Urban Health 2021; 98(2): 259–270. [PubMed: 32888155]

5. Hu L, Li L and Ji J. Machine learning to identify and understand key factors for provider-patient discussions about smoking. Preventive Medicine Reports 2020; 20: 101238. [PubMed: 33224719]

6. Ji J, Hu L, Liu B et al. Identifying and assessing the impact of key neighborhood-level determinants on geographic variation in stroke: a machine learning and multilevel modeling approach. BMC Public Health 2020; 20(1): 1–12. [PubMed: 31898494]

7. Chipman HA, George EI and McCulloch RE. BART: Bayesian additive regression trees. The Annals of Applied Statistics 2010; 4(1): 266–298.

8. Breiman L Random forests. Machine Learning 2001; 45(1): 5–32.

9. Chen T and Guestrin C. XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 785–794.

10. Breiman L, Friedman J, Stone CJ et al. Classification and Regression Trees. Boca Raton, FL: CRC press, 1984.

11. Hothorn T and Zeileis A. partykit: A modular toolkit for recursive partytioning in R. The Journal of Machine Learning Research 2015; 16(1): 3905–3909.

12. Hu L and Gu C. Estimation of causal effects of multiple treatments in healthcare database studies with rare outcomes. Health Services and Outcomes Research Methodology 2021; 21(3): 287–308.

13. Hu L, Ji J and Li F. Estimating heterogeneous survival treatment effect in observational data using machine learning. Statistics in Medicine 2021; In Press.

14. Hu L, Lin JYJ, Sigel K et al. Estimating heterogeneous survival treatment effects of lung cancer screening approaches: A causal machine learning analysis. Annals of Epidemiology 2021; 62: 36–42. [PubMed: 34157399]

15. Hu L, Gu C, Lopez M et al. Estimation of causal effects of multiple treatments in observational studies with a binary outcome. Statistical Methods in Medical Research 2020; 29: 3218–3214. [PubMed: 32450775]

16. Mazumdar M, Lin JYJ, Zhang W et al. Comparison of statistical and machine learning models for healthcare cost data: a simulation study motivated by oncology care model (OCM) data. BMC Health Services Research 2020; 20: 350. [PubMed: 32334595]

17. Hu L, Liu B and Li Y. Ranking sociodemographic, health behavior, prevention, and environmental factors in predicting neighborhood cardiovascular health: a bayesian machine learning approach. Preventive Medicine 2020; 141: 106240. [PubMed: 32860821]

18. Díaz-Uriarte R and De Andres SA. Gene selection and classification of microarray data using random forest. BMC Bioinformatics 2006; 7(1): 3. [PubMed: 16398926]

19. Little RJ, D'Agostino R, Cohen ML et al. The prevention and treatment of missing data in clinical trials. New England Journal of Medicine 2012; 367(14): 1355–1360. [PubMed: 23034025]

20. Sterne JA, White IR, Carlin JB et al. Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. Bmj 2009; 338.

21. NATIONAL RESEARCH COUNCIL. The prevention and treatment of missing data in clinical trials. The National Academies Press, Washington, DC, 2010.

22. Hogan JW, Daniels MJ and Hu L. A bayesian perspective on assessing sensitivity to assumptions about unobserved data. In Molenberghs G, Fitzmaurice G, Kenward MGet al. (eds.) Handbook of Missing Data Methodology., chapter 18. Boca Raton, FL: CRC Press, 2014. pp. 405–434.

23. Hu L, Hogan JW, Mwangi AW et al. Modeling the causal effect of treatment initiation time on survival: Application to HIV/TB co-infection. Biometrics 2018; 74(2): 703–713. [PubMed: 28960243]

24. Tsiatis A Semiparametric theory and missing data. Berlin: Springer, 2007.

25. Garcia RI, Ibrahim JG and Zhu H. Variable selection for regression models with missing data. Statistica Sinica 2010; 20(1): 149. [PubMed: 20336190]

26. Long Q and Johnson BA. Variable selection in the presence of missing data: resampling and imputation. Biostatistics 2015; 16(3): 596–610. [PubMed: 25694614]

27. Wood AM, White IR and Royston P. How should variable selection be performed with multiply imputed data? Statistics in Medicine 2008; 27(17): 3227–3246. [PubMed: 18203127]

28. Meinshausen N and Bühlmann P. Stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 2010; 72(4): 417–473.

29. Kapelner A and Bleich J. Prediction with missing data via bayesian additive regression trees. Canadian Journal of Statistics 2015; 43(2): 224–239.

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

31. Hothorn T, Hornik K and Zeileis A. Unbiased recursive partitioning: A conditional inference framework. Journal of Computational and Graphical statistics 2006; 15(3): 651–674.

32. Efron B Missing data, imputation, and the bootstrap. Journal of the American Statistical Association 1994; 89(426): 463–475.

33. Breiman L Bagging predictors. Machine Learning 1996; 24(2): 123–140.

34. Strobl C, Boulesteix AL, Zeileis A et al. Bias in random forest variable importance measures: Illustrations, sources and a solution. BMC Bioinformatics 2007; 8(1): 25. [PubMed: 17254353]

35. Friedman J, Hastie T, Tibshirani R et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). The Annals of Statistics 2000; 28(2): 337–407.

36. Tibshirani R Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological) 1996; 58(1): 267–288.

37. Friedman J, Hastie T and Tibshirani R. The elements of statistical learning. 2nd ed. New York: Springer series in statistics, 2001.

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

39. Greendale GA, Huang MH, Leung K et al. Dietary phytoestrogen intakes and cognitive function during the menopause transition: results from the SWAN phytoestrogen study. Menopause (New York, NY) 2012; 19(8): 894.

40. Van Buuren S Flexible imputation of missing data. 2nd ed. Boca Raton, FL: Chapman & HallCRC, 2018.

41. Van Buuren S and Groothuis-Oudshoorn CGM. mice: Multivariate imputation by chained equations in R. Journal of Statistical Software 2010; 45(3): 1–67.

42. Stekhoven DJ and Bühlmann P. Missforest—non-parametric missing value imputation for mixed-type data. Bioinformatics 2012; 28(1): 112–118. [PubMed: 22039212]

43. Van Buuren S, Boshuizen HC and Knook DL. Multiple imputation of missing blood pressure covariates in survival analysis. Statistics in Medicine 1999; 18(6): 681–694. [PubMed: 10204197]

44. Waljee AK, Mukherjee A, Singal AG et al. Comparison of imputation methods for missing laboratory data in medicine. BMJ Open 2013; 3(8): e002847.

45. Chen T, He T, Benesty M et al. xgboost: Extreme Gradient Boosting, 2021. R package version 1.4.

46. Friedman J, Hastie T and Tibshirani R. Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software 2010; 33(1): 1. [PubMed: 20808728]

47. Madley-Dowd P, Hughes R, Tilling K et al. The proportion of missing data should not be used to guide decisions on multiple imputation. Journal of Clinical Epidemiology 2019; 110: 63–73. [PubMed: 30878639]

48. Hughes RA, Heron J, Sterne JA et al. Accounting for missing data in statistical analyses: multiple imputation is not always the answer. International Journal of Epidemiology 2019; 48(4): 1294–1304. [PubMed: 30879056]

49. Janssen I, Powell LH, Crawford S et al. Menopause and the metabolic syndrome: the Study of Women's Health Across the Nation. Archives of Internal Medicine 2008; 168(14): 1568–1575. [PubMed: 18663170]

50. Kazlauskaite R, Janssen I, Wilson RS et al. Is midlife metabolic syndrome associated with cognitive function change? The Study of Women's Health Across the Nation. The Journal of Clinical Endocrinology & Metabolism 2020; 105(4): e1093–e1105. [PubMed: 32083676]

51. Carnethon MR, Loria CM, Hill JO et al. Risk factors for the metabolic syndrome: the Coronary Artery Risk Development in Young Adults (CARDIA) study, 1985–2001. Diabetes Care 2004; 27(11): 2707–2715. [PubMed: 15505009]

52. Wei B, Liu Y, Lin X et al. Dietary fiber intake and risk of metabolic syndrome: A meta-analysis of observational studies. Clinical Nutrition 2018; 37(6): 1935–1942. [PubMed: 29137803]

53. Han D, Fang X, Su D et al. Dietary Calcium Intake and the Risk of Metabolic Syndrome: A Systematic Review and Meta-Analysis. Scientific Reports 2019; 9(1): 1–7. [PubMed: 30626917]

54. Feng X, Gao X, Yao Z et al. Low apoA-I is associated with insulin resistance in patients with impaired glucose tolerance: a cross-sectional study. Lipids in Health and Disease 2017; 16(1): 1–7. [PubMed: 28056980]

55. Rao SS, Disraeli P and McGregor T. Impaired glucose tolerance and impaired fasting glucose. American Family Physician 2004; 69(8): 1961–1968. [PubMed: 15117017]

56. Lin JYJ, Hu L, Huang C et al. Strategies for variable selection in large-scale healthcare database studies with missing covariate and outcome data. arXiv preprint 2021; arXiv:2107.09730.

57. Hu L and Hogan JW. Causal comparative effectiveness analysis of dynamic continuous-time treatment initiation rules with sparsely measured outcomes and death. Biometrics 2019; 75(2): 695–707. [PubMed: 30638268]

58. Hu L, Zou J, Gu C et al. A flexible sensitivity analysis approach for unmeasured confounding with multiple treatments and a binary outcome with application to SEER-Medicare lung cancer data. Annals of Applied Statistics 2021; In Press.

59. Ishwaran H and Lu M. Standard errors and confidence intervals for variable importance in random forest regression, classification, and survival. Statistics in Medicine 2019; 38(4): 558–582. [PubMed: 29869423]

60. Williamson BD, Gilbert PB, Carone M et al. Nonparametric variable importance assessment using machine learning techniques. Biometrics 2021; 77(1): 9–22. [PubMed: 33043428]
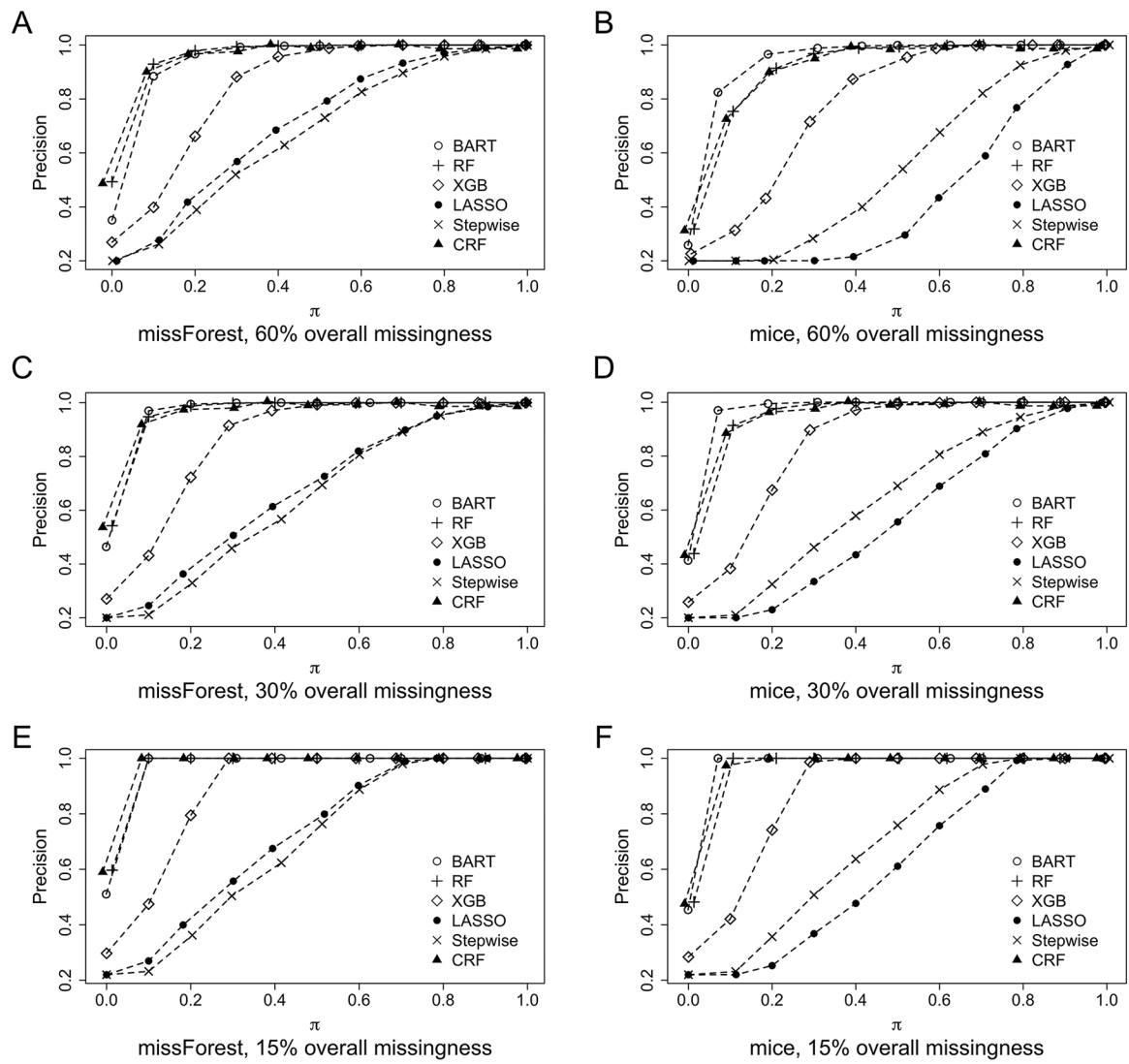
Figure 1.
The precision by threshold values of π for each of six variable selection methods and for sample size $n = 1000$, based on 250 replications. Imputation was performed on 100 bootstrap samples of each replication data set, using two imputation methods `mice` and `missForest` for each of three overall missingness proportions, 15%, 30% and 60%.
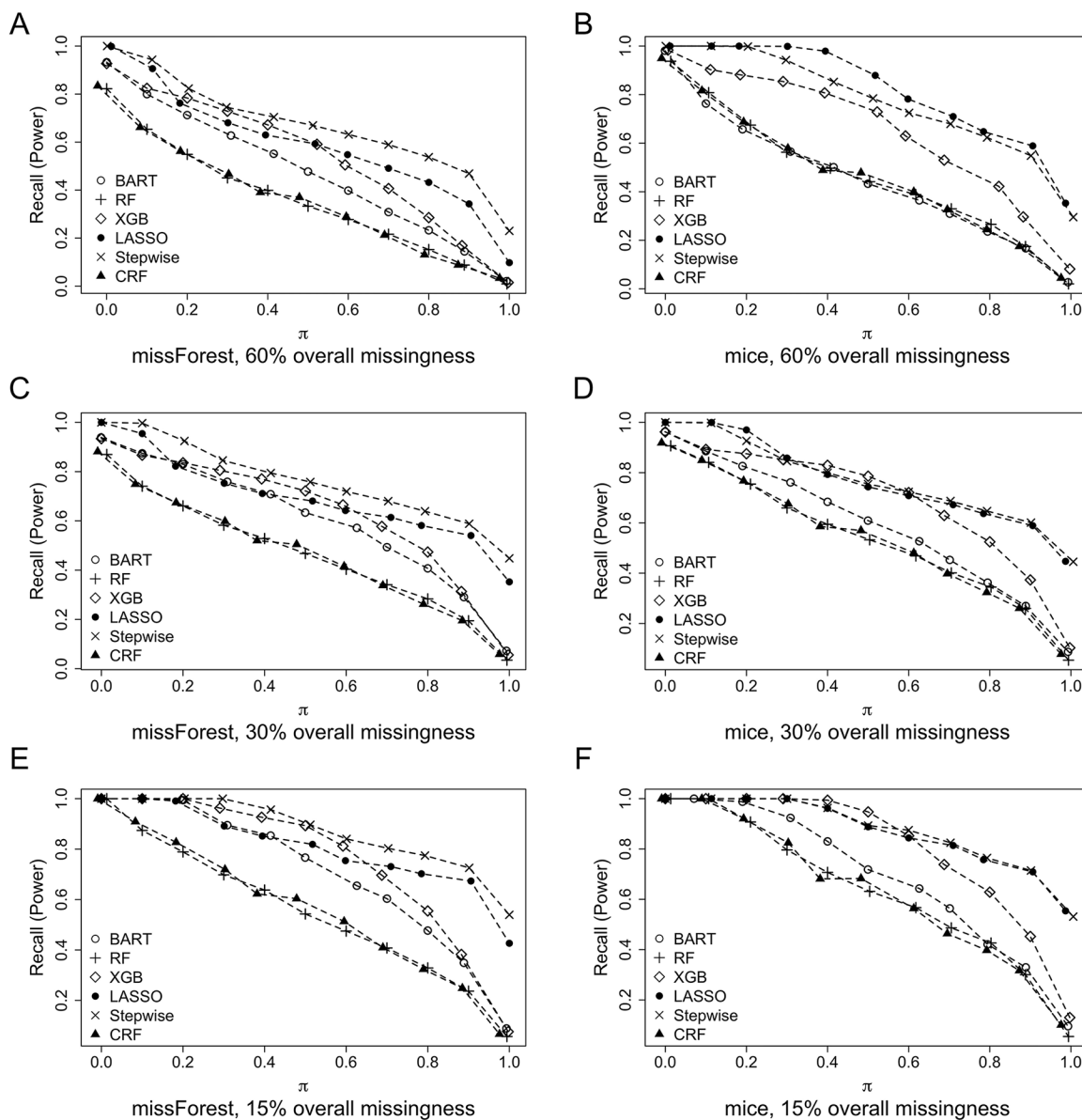
Figure 2.

The recall by threshold values of π for each of six variable selection methods and for sample size $n = 1000$, based on 250 replications. Imputation was performed on 100 bootstrap samples of each replication data set, using two imputation methods `mice` and `missForest` for each of three overall missingness proportions, 15%, 30% and 60%.
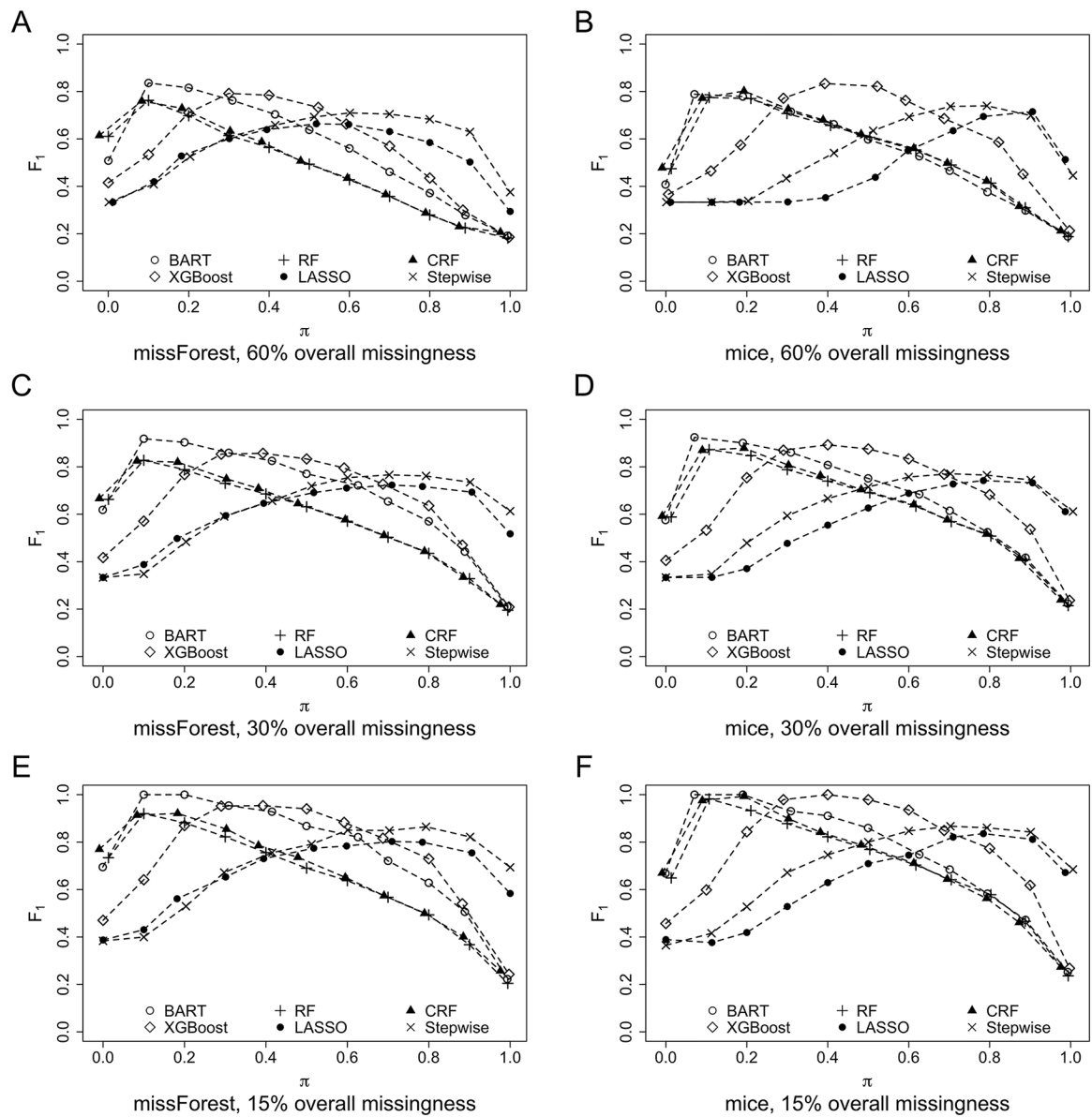
Figure 3.
The F_1 scores by threshold values of π for each of six variable selection methods and for sample size $n = 1000$, based on 250 replications. Imputation was performed on 100 bootstrap samples of each replication data set, using two imputation methods mice and missForest for each of three overall missingness proportions, 15%, 30% and 60%.
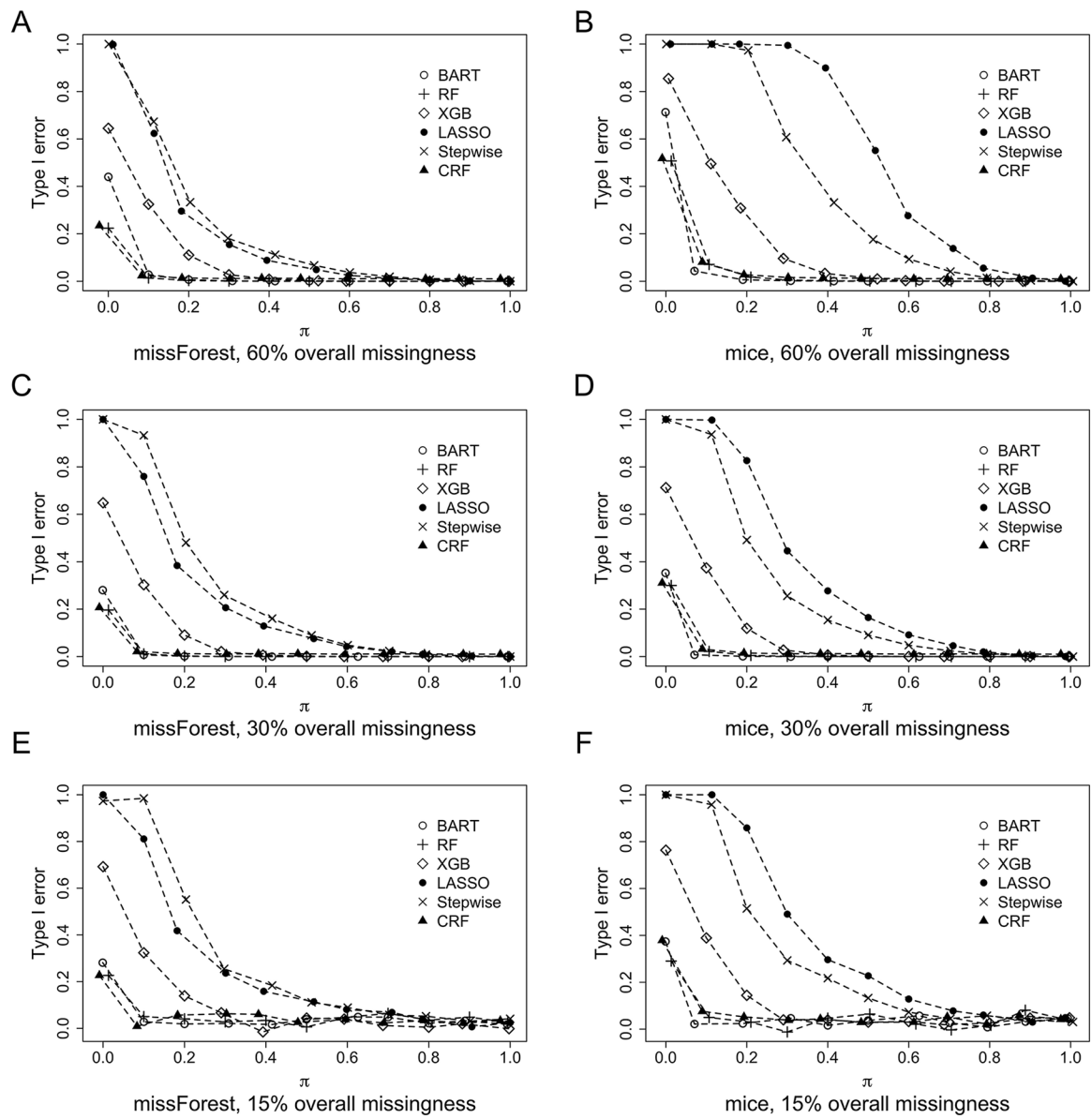
Figure 4.

The type I errors by threshold values of π for each of six variable selection methods and for sample size $n = 1000$, based on 250 replications. Imputation was performed on 100 bootstrap samples of each replication data set, using two imputation methods `mice` and `missForest` for each of three overall missingness proportions, 15%, 30% and 60%.

Figure 5.
Power of each of six methods for selecting each of 10 useful predictors across 250 replications, when the sample size $n = 1000$. Imputation was performed on 100 bootstrap samples of each replication data set, using two imputation methods `mice` and `missForest` for each of three overall missingness proportions, 15%, 30% and 60%.
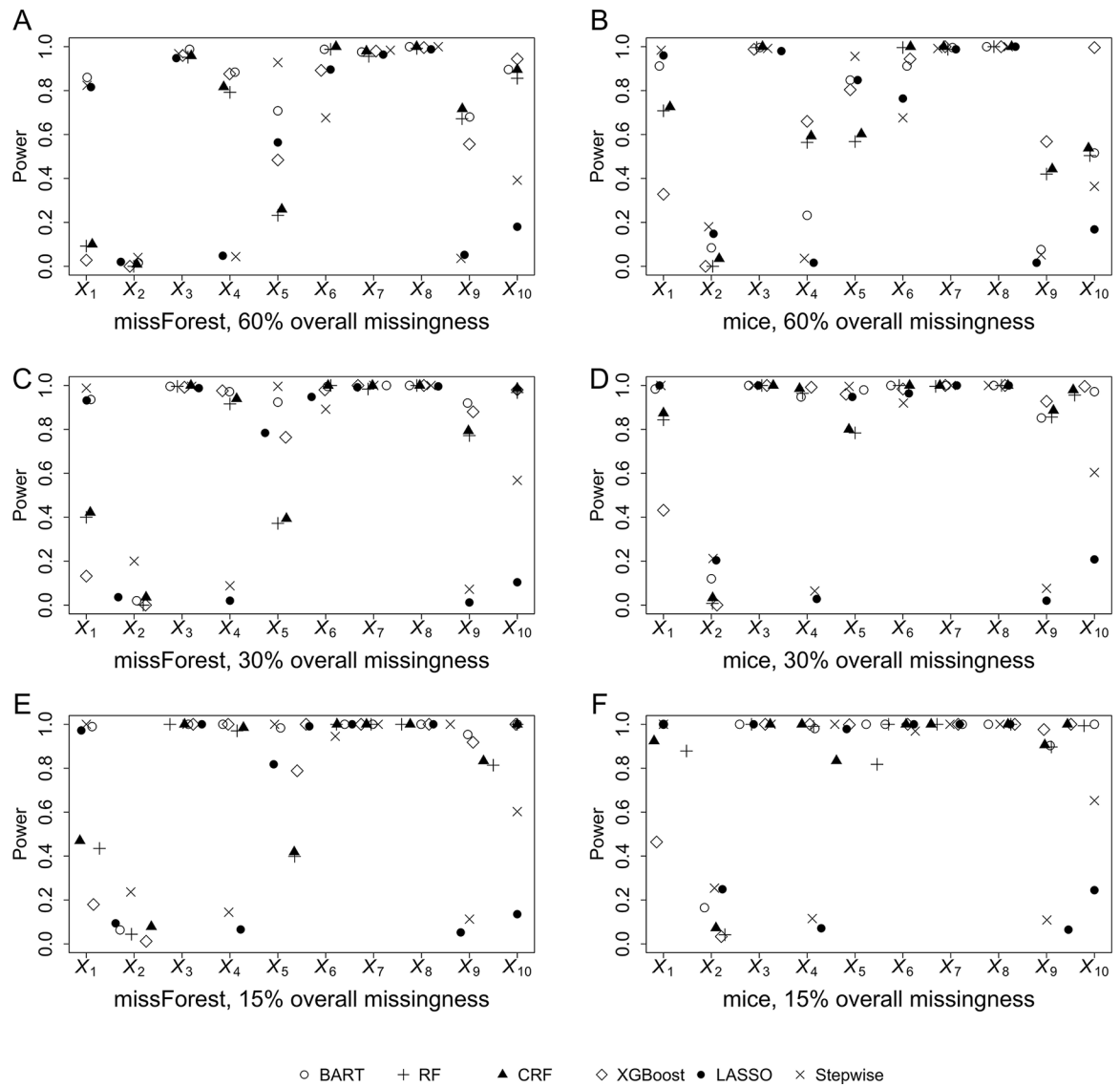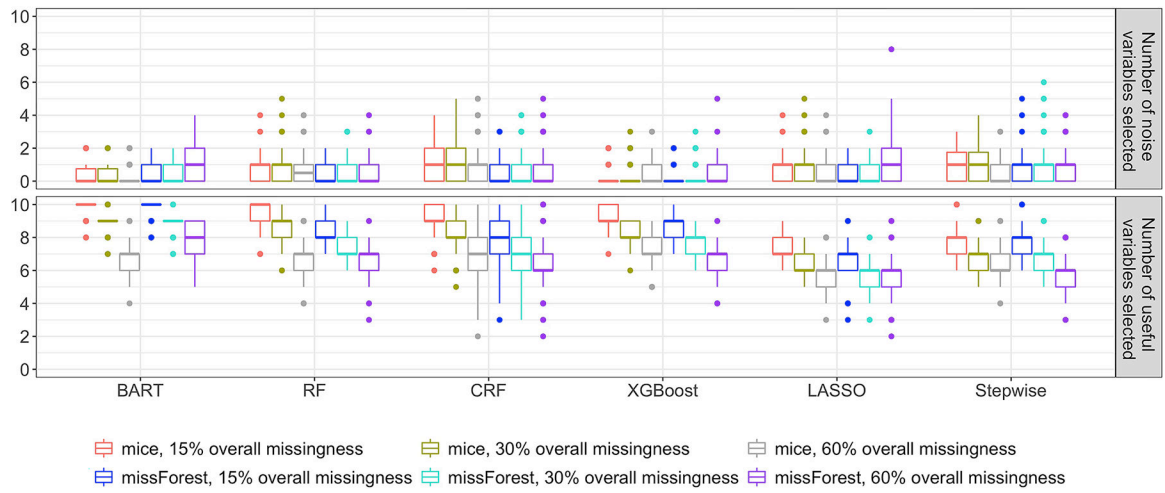
Figure 6.
The distribution of the numbers of selected noise predictors and useful predictors for each of six methods in six scenarios and for *n* = 1000, across 250 replications. The total number of useful predictors is 10 and the total number of noise predictors is 40.
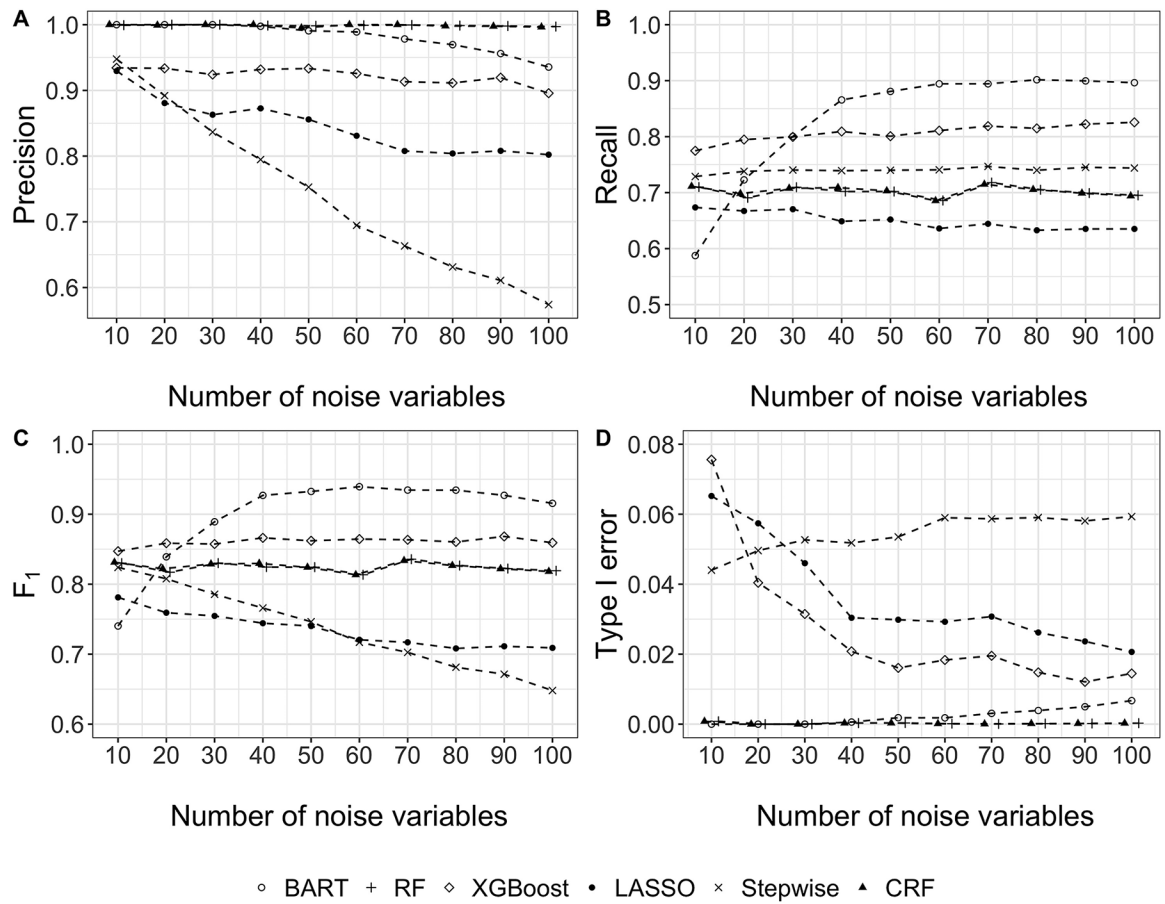
Figure 7.
Precision, recall, F_1 and type I error of each of six methods on complete data with varying number of noise predictors. The total number of useful predictors is 10, and the total number of noise predictors ranges from 10 to 100. Among the noise predictors, 50% are binary variables simulated from Bern(0.5) and 50% are continuous variables generated from N(0,1).
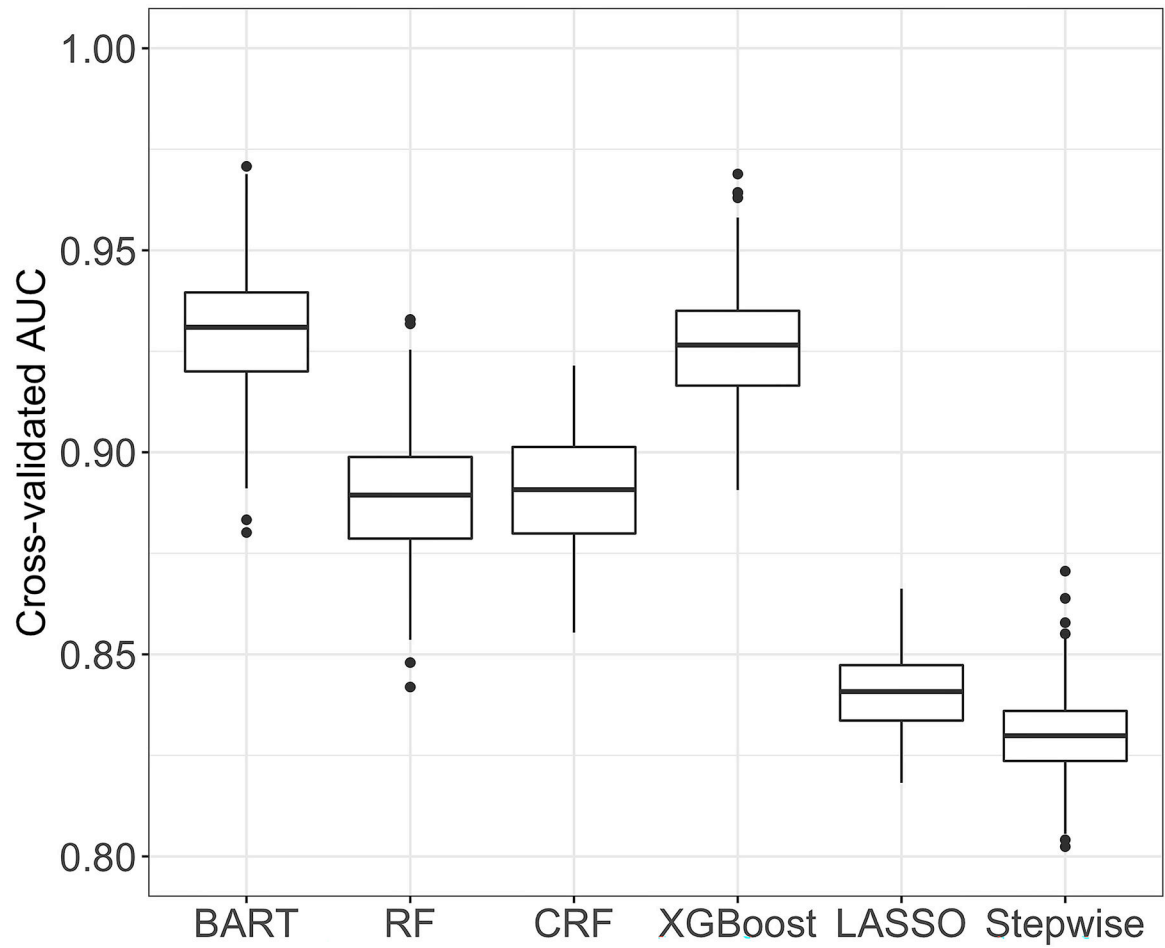
Figure 8.
The distribution of 5-fold cross-validated area under the curve (AUC) among 100 bootstrap samples of the SWAN dataset with imputation performed for each of six methods.

Table 1.

Simulation results for a combination of 6 scenarios = 2 samples sizes ($n = 250$ and $n = 1000$) $\times$ 4 percentages of missingness (complete data, 15% overall missingness, 30% overall missingness and 60% overall missingness). There are a total of 50 predictors, 10 of which are truly related to the response variable and 40 are noise predictors. Imputation was conducted via `mice`. For each of six methods, we show results corresponding to the best threshold values of π (based on F_1) as well as results on the complete cases (CC). Methods are sorted in descending order of F_1 for $n = 1000$.

	$n = 250$					$n = 1000$			
	Precision	**Recall**	F_1	**Type I error**		**Precision**	**Recall**	F_1	**Type I error**
				Complete data					
BART	0.87	0.44	0.58	0.02	BART	1.00	0.87	0.93	0.00
XGBoost	0.82	0.53	0.62	0.04	XGBoost	0.93	0.81	0.86	0.02
CRF	0.95	0.42	0.55	0.01	CRF	1.00	0.72	0.83	0.00
RF	0.94	0.40	0.54	0.01	RF	1.00	0.70	0.82	0.00
Stepwise	0.68	0.56	0.60	0.08	Stepwise	0.79	0.74	0.76	0.05
lasso	0.83	0.36	0.51	0.03	lasso	0.87	0.65	0.74	0.03
				15% missingness in Y and 30% overall missingness					
BART $\pi = 0.2$	0.64	0.55	0.59	0.08	BART $\pi = 0.1$	0.97	0.89	0.92	0.01
XGBoost $\pi = 0.3$	0.77	0.59	0.66	0.05	XGBoost $\pi = 0.4$	0.97	0.83	0.89	0.01
RF $\pi = 0.2$	0.76	0.52	0.62	0.04	RF $\pi = 0.1$	0.91	0.84	0.87	0.02
CRF $\pi = 0.2$	0.77	0.53	0.63	0.03	CRF $\pi = 0.2$	0.98	0.77	0.86	0.00
Stepwise $\pi = 0.5$	0.60	0.55	0.56	0.11	Stepwise $\pi = 0.7$	0.89	0.69	0.77	0.02
lasso $\pi = 0.7$	0.73	0.43	0.52	0.05	lasso $\pi = 0.8$	0.90	0.64	0.74	0.02
BART CC	0.56	0.47	0.51	0.10	BART CC	0.93	0.79	0.85	0.02
XGBoost CC	0.82	0.44	0.56	0.05	XGBoost CC	0.87	0.77	0.81	0.04
CRF CC	0.44	0.56	0.48	0.13	CRF CC	0.82	0.76	0.79	0.03
RF CC	0.43	0.55	0.47	0.15	RF CC	0.84	0.75	0.78	0.04
Stepwise CC	0.78	0.29	0.42	0.02	Stepwise CC	0.80	0.62	0.71	0.04
lasso CC	0.88	0.23	0.36	0.03	lasso CC	0.80	0.60	0.70	0.04

Table 2.

Variable selection results by each of six methods, with the best imputation method and threshold value of π suggested in simulations. BART was used with `missForest` and $\pi = 0.1$, CRF and RF with `mice` and $\pi = 0.2$, XGBoost with `mice` and $\pi = 0.5$, lasso with mice and $\pi = 0.9$, Stepwise with `mice` and $\pi = 0.8$. Definitions of the variable names appear in Web Table 2.

Methods	# selected variables	Selected variables
BART	17	APOARES, BMI, BP, DIABP, DTTLIN, GLUCOSE, LPA, RACE, RESTLES, SHBG, SPBMD, SYSBP, T, TPA, TRIGRES, WAIST, WHRATIO
XGBoost	18	APOARES, BMI, DIABP, EDUCATION, E2AVE, GLUCOSE, HPBMD, INSULIN, LMPDAY, LPA, PAI1, SHBG, SPBMD, SYSBP, TPA, TRIGRES, WAIST, WHRATIO
CRF	11	BMI, DIABP, EDUCATION, INSULIN, LPA, PAI1, SYSBP, TPA, TRIGRES, WAIST, WHRATIO
RF	12	BMI, CRP, DIABP, EDUCATION, INSULIN, LPA, PAI1, SYSBP, TPA, TRIGRES, WAIST, WHRATIO
Stepwise	10	BMI, DIABP, EDUCATION, LPA, NOINSURE, PHYSWORK, RESTLES, SPORTS, SYSBP, TRIGRES
lasso	10	ALLCALC, APOARES, DIABP, DTTSWET, E2AVE, GLUCOSE, LPA, SYSBP, TRIGRES, WAIST