Austin Biometrics and Biostatistics

Open Access

Austin Publishing Group

Research Article

Parallelization Using Different Strategies and its Influence on Random-Forest-Based Statistical Methods: A Case Study on Iterative Missing Data Imputation

Hong S, Sun Y, Li H and Lynn HS*
Department of Biostatistics, Fudan University, China

***Corresponding author:** Lynn HS, Department of Biostatistics, School of Public Health, Key Laboratory on Public Health Safety of the Ministry of Education, Fudan University, Shanghai, China

Received: December 08, 2020; **Accepted:** February 24, 2021; **Published:** March 03, 2021

Abstract

Random forest has proven to be a successful machine learning method, but it also can be time-consuming for handling large datasets, especially for doing iterative tasks. Machine learning iterative imputation methods have been well accepted by researchers for imputing missing data, but such methods can be more time-consuming than standard imputation methods. To overcome this drawback, different parallel computing strategies have been proposed but their impact on imputation results and subsequent statistical analyses are relatively unknown. Newly proposed random forest implementations, such as ranger and randomForestSRC, have provided alternatives for easier parallelization, but their validity for doing iterative imputation are still unclear. Using random-forest imputation algorithm missForest as an example, this study examines two parallelized methods using newly proposed random forest implementations in comparison with the two parallel strategies (variable-wise distributed computation and model-wise distributed computation) using language-level parallelization from the software package. Results from the simulation experiments showed that the parallel strategies could influence both the imputation process and the final imputation results differently. Different parallel strategies can improve computational speed to a variable extent, and based on simulations, ranger can provide performance boost for datasets of different sizes with reasonable accuracy. Specifically, even though different strategies can produce similar normalized root mean squared prediction errors, the variable-wise distributed strategy led to additional biases when estimating the mean and inter-correlation of the covariates and their regression coefficients. And parallelization by randomForestSRC can lead to changes in both prediction errors and estimates.

Keywords: Random forest; Parallel computation; Missing data iterative imputation

Abbreviations

OOB: Out-of-Bag; RF: Random Forest; MCAR: Missing Completely at Random; MAR: Missing at Random

Introduction

Random forest has proven to be a successful machine learning method with successful applications [1]. As missing data are common in most research, various kinds of imputation methods have been proposed for handling missing data problems. Stekhoven and Buhlmann [2] proposed the missForest algorithm based on a Random Forest (RF) machine learning method [3], and it has been used in different studies and benchmarked against other imputation methods [4-7]. MissForest has been shown to have superior predictive accuracy under certain circumstances, but it necessitates the building of a large number (default is 100) of trees during the imputation process for a single variable per iteration and usually several iterations are required. Likewise, missForest can be computationally intensive and time-consuming for large datasets, thereby limiting its usability. Moreover, various multiple imputation methods have been proposed based on random forests [8], and they all do iterative computations.

To boost performance, two parallel computing strategies (referred to as "forests" and "variables" in the software package) suitable for "long" and "wide" datasets, respectively [9], were implemented in missForest with the release of version 1.4. However, there has not been any published evaluation of their difference on predictive accuracy and subsequent impact on statistical analyses. The implicit assumption is that these two strategies are equally valid and will lead to similar results. Recently, new implementations of the RF algorithm, like ranger [10] and randomForestSRC [11] software packages, have been proposed to provide new functionalities and computational speed improvements. But applications of such implementations in doing iterative prediction tasks are rarely discussed and their influences on the results are still unknown.

Using missForest algorithm as an example, the algorithm was reimplemented using the two recently proposed RF software packages, and this study uses simulation experiments to address the differences in both imputation accuracy and computation efficiency among these parallel computation strategies of missForest. Computational efficiency can be critical for handling large datasets; thus, this study's results can be of use to both data analytics practitioners and

Citation: Hong S, Sun Y, Li H and Lynn HS. Parallelization Using Different Strategies and its Influence on Random-Forest-Based Statistical Methods: A Case Study on Iterative Missing Data Imputation. Austin Biom and Biostat. 2021; 6(1): 1037.

methodologists for imputation methods.

Materials and Methods

MissForest algorithm

In missForest, the variables containing missing values are initialized for imputation by replacing the missing cells by corresponding mean values (for continuous variables), or by the most frequent category (for categorical variables). A variable under imputation is then divided into two distinct parts: the observed part that contains no missing values, and the missing part that serves as the prediction set. A random forest is fitted using the observed part as response and the corresponding values of the other variables as predictors, and the missing part is replaced with the predicted values from the random forest. The algorithm then proceeds to the next variable to be imputed, and the iteration stops when the difference between the current and previously imputed values increase or if the maximum number iteration is reached. Since the release of missForest version 1.4, two parallel strategies have been implemented to increase the computational efficiency when applying random forest imputation to large datasets.

Strategy 1: distributing the computation of imputing a single variable: In the first strategy, the building of the ensemble of trees for a variable to be imputed are divided into smaller subsets and distributed into different computing processes based on the number of core processors in the computer. The results from different ensembles of smaller trees are recombined into a single one, and the final predictions are derived from the combined ensemble of trees. Each variable to be imputed undergoes this process until all the variables have been imputed in a single iteration. This strategy is most useful if the process of building a random forest is time-consuming and the number of variables in the dataset is relatively small.

Strategy 2: distributing the computation of different variables: In the second strategy, the computation of the random forest for each variable to be imputed in a single iteration is distributed to different computing processes. The imputations of the variables are done simultaneously and independent of each other with the building of the ensemble of trees for each variable performed by a single process. After all the variables have been imputed, the results are recombined to form a single complete dataset. The current iteration is then finished, and the algorithm moves to the next iteration. This strategy can be useful for datasets containing many variables while the time consumption for building the random forest for a single variable is small.

Accelerated random forests

The R software packages, ranger and randomForestSRC, have extended the original random forest algorithms in different ways and they both provided parallelized implementations of the random forest algorithm. The ranger software package is fast implementation of random forest, particularly suited for high dimensional data. And the "rfsrc.fast" function in randomForestSRC provides fast approximate random forests. Both software packages support classification, regression, and survival forests. The missForest algorithm was reimplemented using ranger and randomForestSRC with parameters adjusted to eliminate differences in sampling processes, as the "rfsrc.fast" function in randomForestSRC does not do bootstrap sampling by default. The source code for self-written software package used in this study can be found online [12].

Simulation studies

To further investigate the influence of the choice of parallel strategies on imputation, a series of simulations and analyses were carried out using R, version 3.6 (R Core Team, Vienna, Austria) [13]. Four sequential stages were involved:

- **Data generation:** Complete datasets were simulated based on pre-defined scenarios.
- **Amputation:** The complete datasets were made incomplete based on specified rules.
- **Imputation:** the missing values contained in the simulated datasets were filled in by missForest using different parallel strategies.
- **Analysis:** Statistical analysis were performed on both the original complete datasets and the corresponding imputed datasets, and comparisons were made.

The computational time costs of different strategies were also compared based on simulations of missing completely at random (MCAR) data containing 50% missing data cells on a laptop computer with a multi-core CPU installed to demonstrate the performance gain using different parallel strategies.

Data generation: The data structures were made as simple as possible with a response Y and just two covariates X_1 and X_2 to enhance the investigation of the influence of the two parallel strategies on imputation results. Also, a large variance was used to get more discriminative results. Three different sets of 2000 simulated datasets containing 200 observations each were generated based on following settings:

Uncorrelated covariates with linearly dependent response:

$$\begin{bmatrix} Y \\ X_1 \\ X_2 \end{bmatrix} \sim Normal\left(\begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 21 & 10 & 10 \\ 10 & 10 & 0 \\ 10 & 0 & 10 \end{bmatrix}\right),$$

Which gives rise to the linear regression model

$Y=\beta_1 X_1+\beta_2 X_2+\varepsilon$, where $\varepsilon \sim$ Normal (0, 1), $\beta_1=\beta_2=1$.

And the conditional distribution [14] of Y given $X_1=x_1$ and $X_2=x_2$ is

$(Y|X_1 = x_1, X_2 = x_2) \sim$Normal $(x_1 + x_2, 1)$.

Correlated multivariate normal data:

$$\begin{bmatrix} Y \\ X_1 \\ X_2 \end{bmatrix} \sim Normal\left(\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 10 & 10\rho & 10\rho \\ 10\rho & 10 & 10\rho \\ 10\rho & 10\rho & 10 \end{bmatrix}\right)$$

The correlation coefficients were ρ=0.25 or ρ=0.75, roughly corresponding to weakly correlated and strongly correlated data. This multivariate distribution leads to the following conditional distributions of Y given $X_1 = x_1$ and $X_2 = x_2$:

$(Y|X_1 = x_1, X_2 = x_2, X_2=x_2) \sim$Normal $(0.2x_1+0.2x_2+0.6, 9)$, and

$$(Y|X_1 = x_1, X_2 = x_2) \sim \text{Normal}\left(\frac{3}{7}x_1 + \frac{3}{7}x_2 + \frac{1}{7}, \frac{25}{7}\right).$$

Altogether, the simulation consists of three different data generation scenarios: (1) multivariate normal data with independent covariates; (2) moderately correlated multivariate normal data; and (3) strongly correlated multivariate normal data.

Amputation: Amputation functions [15] provided by the "MICE" [16] R package were used in this study to generate missing values. Missing at random (MAR) patterns were introduced by setting X_1 and/or X_2 to be missing depending on Y. Specifically, the probability of each observation being missing was set to 50% according to a standard right-tailed logistic function on Y; thus the probability of the covariates being missing is higher for observations with higher values of Y. Two MAR patterns are generated, whereby either both covariates are missing (i.e., two missing cells) or only one of the covariates is missing (i.e., one missing cell).

Imputation: The amputed datasets underwent imputation by missForest, and default parameter values (number of trees grown was set to 100, and maximum iteration was set to 10) were accepted as recommended by the original article [2]. The number of distributed computing processes was set to three, which equals to the number of variables in the dataset (the maximum allowed by missForest), to allow for more computing resources available for "forests" strategy. Imputation without parallelization, parallelized imputation by forests and by variables were performed.

Analysis: Comparisons were made between the two parallel strategies, along with the original sequential algorithm, based on:

- The number of iterations performed using different parallel strategy.
- Relative bias for the mean and for the standard deviation of the imputed variable,

$$\frac{mean(V_{imp})}{mean(V_{true})} - 1, \text{ and } \frac{sd(V_{imp})}{sd(V_{true})} - 1$$

where V is either one of the imputed variables (X_1 or X_2), V_{true} is the original vector of true values, V_{imp} is the data vector after imputation, and the mean and standard deviation are computed over all the data values;

- The relative bias of the coefficient estimate, $(\beta_p\text{-}\beta_p)/\beta_p$, p = 1, 2 or 3, corresponding to the intercept (if any), X_1 or X_2;
- Normalized root mean squared error (NRMSE) values,

$$\sqrt{\frac{mean((X_{true} - X_{imp})^2)}{\text{var}(X_{true})}},$$

where X_{true} and X_{imp} are the true and imputed data matrix, respectively, and the mean and variance are computed only over the missing values.

Pearson correlation coefficients were also estimated for certain data scenarios when investigating the influence of imputation on the relationships between imputed variables. If the two parallel algorithms are equivalent and valid, then their imputation results should not be dissimilar with the sequential algorithm in imputation accuracy for all four criteria.

Results

The results from three different parallel strategies showed variations in iteration numbers, relative bias of sample mean, relative differences of standard derivation and regression estimates in linear regression data scenario. However, such differences showed correlated relationship with different data scenarios.

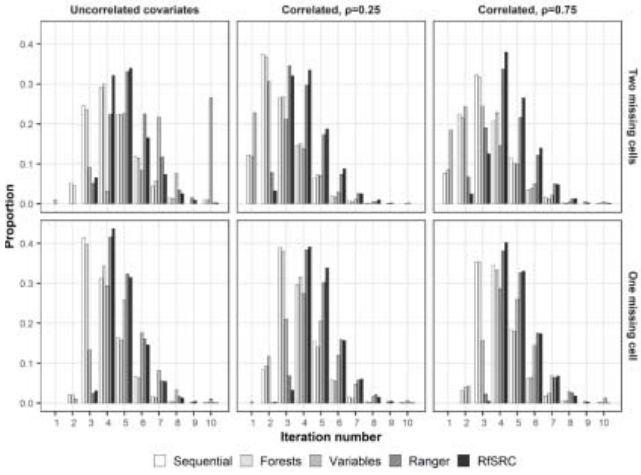

Figure 1: Number of iterations for different parallel modes in different data generation settings and missing patterns.

Iterations performed

The number of iterations of imputation with the "variables" parallel strategy was very different ($p<0.001$ across all scenarios, Fisher's exact test) from the other two strategies for all eight data scenarios, while sequential imputation and parallel "forests" strategies were more similar. For the parallel "forests" and sequential strategies, most imputation runs stopped at two to four iterations with only a small number of runs (<0.25% overall) reaching the maximum number of ten loops. For the "variables" parallel imputation, however, imputation often require larger number of iterations even for data with one missing cell per observation (median no. of iterations=3, 4 for two missing cells, one missing cell, respectively). Note also that with two missing cells per observation, an exceedingly large proportion of runs (26.5%) stopped at the maximum iteration number for the "variables" strategy. For ranger and randomForestSRC implementations, most runs stopped at 4 or 5 iterations, and showed different patterns compared with original missForest implementations (Figure 1).

Relative bias of sample mean

The "variables" parallel imputation strategy resulted in more biased mean estimates in datasets with multiple missing cells per observation. With two missing cells per observation for the "uncorrelated covariates" scenario, the "variables" strategy had an additional downward relative bias when estimating the mean of X_1 (median=-9.8%) compared with the sequential (median=-6.1%, $p<0.001$) and "forests" (median=-5.5%, $p<0.001$) strategies, while for X_2 an additional upward relative bias was introduced (median=-22.8%, -22.8%, -9.7% for "sequential", "forests", "variables", respectively). For weakly correlated data, the sample mean of X_1 was similar (median = -6.9%, -6.7%, -6.5%), but for X_2 a downward bias was introduced by "variables" (median=-6.3%, -6.3%, -10.0%). For strongly correlated data, the "variables" strategy produced biased downward sample means of X_1 (median=-13.5%, -13.2%,

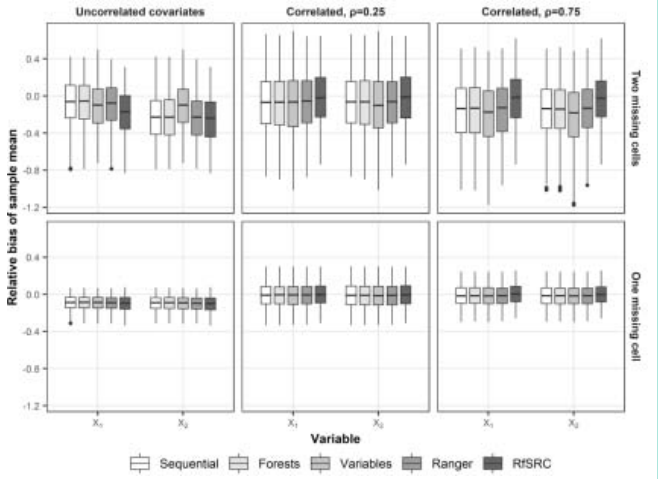

Figure 2: Relative bias of the estimated means of variables.

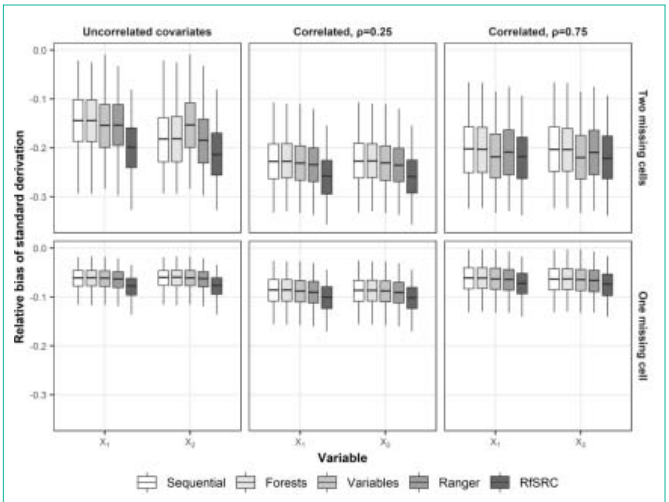

Figure 3: Relative bias of standard derivations of variables.

-17.5%), as well as X_2 (median -13.7%,-14.2%, -18.2%). For ranger implementation, results were similar with sequential missForest, but for randomForestSRC implementation, differences can be observed for X_1 in "uncorrelated covariates" scenario (-6.1% and -17.1%for "sequential" and "randomForestSRC" respectively), and upward bias for weakly correlated data (-6.9% and -2.0%) and strong correlated data (-13.5%and -1.5%).Similar patterns can be observed for X_2 in weakly and strongly correlated data. When there was only one missing cell per observation, the relative bias of the sample mean was similar across all the strategies (Figure 2).

Relative bias of standard derivation

For estimating the standard derivation, all three strategies were systematically biased downward. The results of the sequential and "forests" strategies were similar, with the "variables" strategy yielding slightly more biased estimates (Figure 3). For example, with two missing cells per observation, the median relative biases for X_1 and X_2 were -14.4%, -14.4%, -15.4% and -18.2%, -18.1%, -15.3% for "sequential", "forests", "variables", respectively, for the "uncorrelated covariates" scenario. While for strongly correlated data, the median relative biases for X_1 and X_2 were -20.2%, -20.3%, -21.9%, and -20.3%,

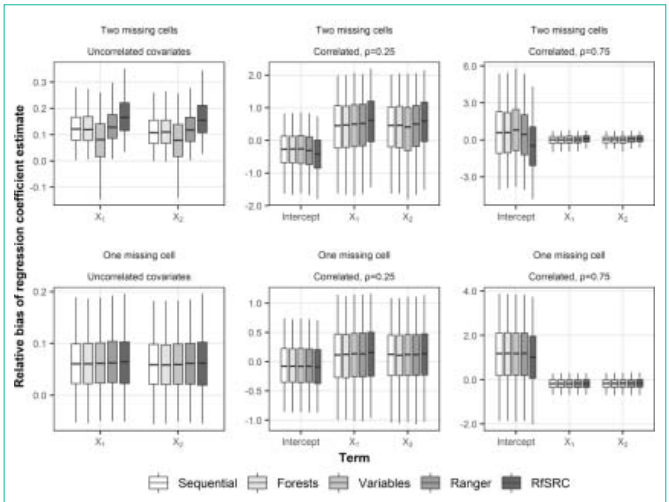

Figure 4: Relative bias of the estimated regression coefficients.

-20.3%, -22.0%, respectively. However, implementation-using randomForestSRC can lead to consistent underestimated standard deviation even for data with one missing cell per observation, while implementation-using ranger can yield similar results with sequential missForest (Figure 3).

Relative bias of regression coefficient estimates

MissForest led to biased regression coefficient estimates when covariates are outcome-dependent MAR, and the "variables" parallel strategy can cause additional bias. With two missing cells per observation, the "sequential" and "forests" strategies were similar for the "uncorrelated covariates" scenario, but the "variables" strategy produced additional downward relative bias (X_1: median=12.1%, 12.0%, 8.1%, for "sequential", "forests", "variables", respectively; X_2: median=10.7%, 11.0%, 7.9%, respectively).For weakly correlated data, the median relative biases for coefficient estimates of (intercept, X_1, X_2) were (-16.4%, -15.8%, -15.5%), (9.0%, 9.4%, 9.9%), and (9.1%, 9.2%, 8.2%) for the "sequential", "forests", and "variables" strategies, respectively. While for strongly correlated data, the median relative biases for coefficient estimates of (intercept X_1, X_2) were (8.3%, 8.5%, 11.5%),(0.2%, 0.4%, 1.1%) and (1.3%, 1.2%, 0.9%), respectively. For randomForestSRC implementation, upward relative bias (X_1: median = 12.1%, 16.6%, p<0.001 for "sequential", randomForestSRC respectively; X_2: median = 10.7%, 15.5%, respectively, $p < 0.001$) can be observed for the "uncorrelated covariates" scenario. Also, downward bias can be observed for the estimated intercept in strongly correlated data (median = 8.3%, -6.6%, $p < 0.001$). For datasets with only one missing cell per observation, imputation using different parallel strategies gave similar results (Figure 4).

Bias of correlation between covariates

The choice of strategy influenced the correlation between the imputed covariates. With two missing cells per observation, all three strategies produced inflated correlations between X_1 and X_2 for the "uncorrelated covariates" scenario (Figure 5) but the "variables" parallel strategy resulted in the most biased correlation estimates (median = 0.18 compared to 0.12, for both "sequential" and "forests" strategies). For weakly correlated data, the correlation coefficients were similar, but for highly correlated data the correlations were

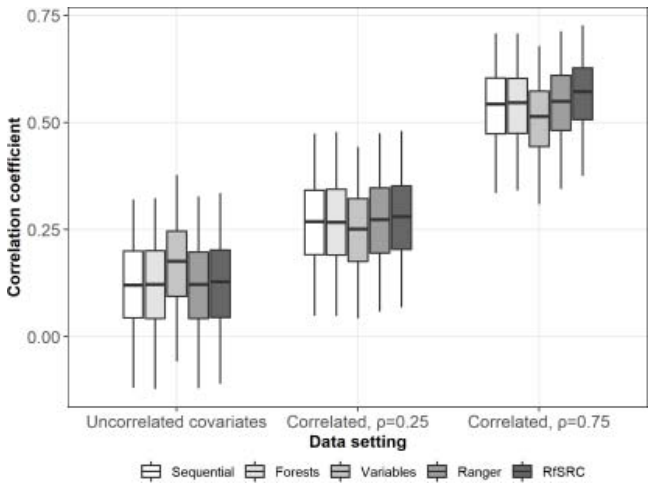

Figure 5: Bias of correlation between covariates.

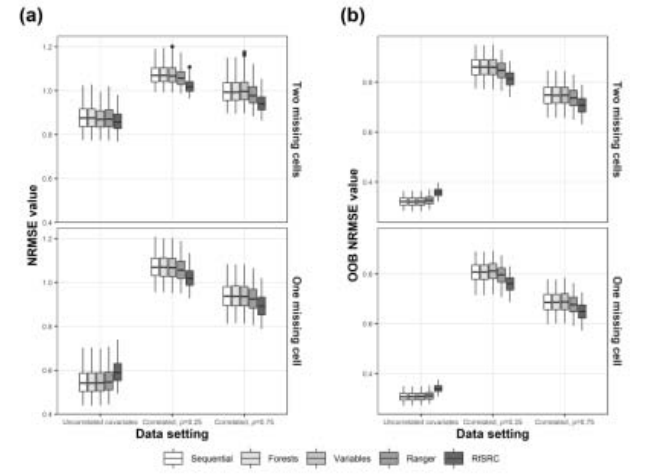

Figure 6: Normalized root mean squared errors (NRMSE) in final imputed data. a) NRMSE calculated based on true data; b) Out-of-bag (OOB) errors calculated from OOB observations (Uncorrelated: Uncorrelated covariates).

biased downward with the "variables" strategy yielding the lowest estimate (median = 0.51 compared to 0.55 and 0.54 for "sequential" and "forests", respectively) and randomForestSRC yielding the highest estimate (median = 0.57). For other circumstances, both implementations using ranger and randomForestSRC can yield similar results as "sequential" missForest.

NRMSE values

The NRMSE values for all three strategies appeared similar except for implementation using randomForestSRC, and the change can be upward and downward depending on the data scenarios, regardless of whether they were calculated based on the original data (Figure 6a), or from the Out-of-Bag (OOB) values (Figure 6b).

Time cost

From the time cost of different strategies, it can be concluded all parallel strategies can reduce the time cost for data sets of different sizes except for randomForestSRC can lead to even larger time consumption for datasets with small variable or observation

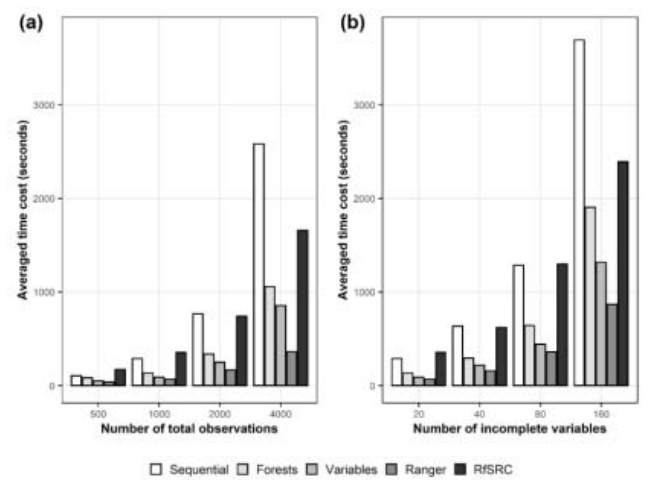

Figure 7: Averaged time cost for different parallel strategies when performing imputation for datasets of different sizes. a) 20 variables and varying observation numbers; b) 1000 observations and varying variable numbers.

numbers. Specially, the ranger implementation can be nearly 5 times faster than the sequential missForest on a laptop computer and can run the fastest across all scenarios (Figure 7).

Discussion

This study examines different parallel strategies of the RF-based imputation method missForest and documents for the first time their influence on imputation results. By distributing imputation computation to multiple computing processes, reduction in time consumption can be achieved with multi-core processors. For missForest, by using functionalities from software packages for parallel computation support, in the "forests" parallel strategy, the imputation for a single variable was parallelized, and in the "variables" mode, the single iteration of imputation for different variables was parallelized. In implementations using ranger and randomForestSRC, the parallel computation will be handled within the software package automatically and is transparent to researchers.

Depending on the data structures, our findings indicate that the "variables" strategy can lead to variations in the final imputation results compared with the original missForest "sequential" algorithm. The "variables" strategy yielded additional upward or downward biases when estimating covariate means, correlation between covariates, and regression coefficients. This can harm reproducibility and may even lead to false inference. Moreover, the little variation in NRMSE values between the different strategies may give a false sense of consistency between them. This also highlights the fact that evaluating imputation results based solely on NRMSE values can lead to unreliable conclusions.

For implementation using randomForestSRC, it can cause changes in nearly all analysis aspects even NRMSE values, and the directions of such changes varied with the data used so it cannot be easily predicted. Researchers should be cautious when using randomForestSRC for doing iterative tasks. From the results of this study, it can be concluded that performance boost and accuracy can be balanced for random-forest-based algorithms using ranger software package.

The difference in results between the two parallel strategies implemented in missForest is a consequence of their different computing processes. In the parallel "forests" strategy, the imputation of the current variable is based on the latest state of the imputation dataset, and the observations of the previously imputed variable are updated before the start of the imputation of the current variable. The computation of a single tree in an ensemble is also done independent of other trees, so this parallel strategy should be similar to the "sequential" strategy that computes all the trees in an ensemble using a single processor rather than multiple processors. On the other hand, in the parallel "variables" strategy the imputation of different variables is done in parallel such that their computation is based on the same previously updated imputed dataset rather than variable-wise sequentially updated imputed datasets. This implies that imputed results are not updated until one cycle of imputation is finished for all the variables imputed in parallel. Therefore, the imputation of the variables within a single iteration can no longer be considered sequential, resulting in different final imputed values from the "sequential" strategy. For implementations using ranger and randomForestSRC, related technical details can be found in the websites of the software packages and will not be further discussed in this paper.

The simulations in this study focused on "long" data where the number of observations is larger than the number of variables in the dataset. However, for "wide" data with large number of variables, the impact of the non-sequential updating of imputed values in the "variables" parallel strategy can be even larger, especially when missing values are scattered across multiple variables with low inter-correlations. It should be noted that the data settings in this study were designed to accentuate the differences and consequences of the two different parallel strategies. In practice, however, datasets like the simulated data in this study may not be suited for parallel computation as they are not large enough in terms of number of variables or observations. Also, the parallelization algorithm can lead to additional time cost, resulting in more computational time than expected for certain datasets.

This study highlights the importance of thorough testing of computational algorithms. In particular, it is the lack of technical details in the official missForest documentations that prompted this investigation. Machine learning methods like random forests are computationally intensive. Likewise, their application to big data problems will necessitate the use of parallel computation algorithms, but developers and users of such statistical software may be wise to devise simple simulation experiments to test and compare the algorithms before using them for data analyses. Finally, although we focused on the missForest method the lessons learned here is not peculiar to it, and other iterative imputation methods (e.g., MICE) may be faced with similar problems when adapted for parallel computation.

Conclusion

The problem of using parallel computation has been brought into the forefront with this study's investigation of the two parallel strategies implemented in missForest and two proposed strategies using newly implemented software packages for random-forest. It is expected that the proliferation of large datasets and complex computational methods will continue to fuel the use of parallel algorithms. The careful analysis of these algorithms is therefore especially important, and the documentation of these algorithms should include sufficient technical details and test experiments to inform researchers of potential problems. Based on results of this study, the ranger software package is recommended for performing random-forest modeling, especially for iterative tasks.

References

1. Biau G, Scornet E. A random forest guided tour. Test. 2016; 25: 197-227.
2. Stekhoven DJ, Buhlmann P. MissForest--non-parametric missing value imputation for mixed-type data. Bioinformatics. 2012; 28: 112-118.
3. Liaw A, Wiener M. Classification and regression by random Forest. R news. 2002; 2: 18-22.
4. Waljee AK, Mukherjee A, Singal AG, Zhang Y, Warren J, Balis U, et al. Comparison of imputation methods for missing laboratory data in medicine. BMJ Open. 2013; 3: e002847.
5. Shah AD, Bartlett JW, Carpenter J, Nicholas O, Hemingway H. Comparison of random forest and parametric imputation models for imputing missing data using MICE: a CALIBER study. American Journal of Epidemiology. 2014; 179: 764-774.
6. Tang F, Ishwaran H. Random Forest Missing Data Algorithms. Statistical Analysis and Data Mining. 2017; 10: 363-377.
7. Ramosaj B, Pauly M. Predicting missing values: a comparative study on non-parametric approaches for imputation. Computational Statistics. 2019; 34: 1741-1764.
8. Mayer M. missRanger: Fast Imputation of Missing Values. 2019.
9. Stekhoven DJ. missForest: Nonparametric Missing Value Imputation using Random Forest. 2013.
10. Wright MN, Ziegler A. Ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R. Journal of Statistical Software. 2017; 77: 1-17.
11. Ishwaran H, Kogalur UB, Blackstone EH, Lauer MS. Random survival forests. The Annals of Applied Statistics. 2008; 2: 841-860.
12. Hong S. missForestFast. 2020.
13. R Core Team. R: A language and environment for statistical computing. 2020.
14. Searle SR, Gruber MHJ. Linear Models. 2nd ed. Hoboken, New Jersey: John Wiley & Sons. 2016: 67-68.
15. Schouten RM, Lugtig P, Vink G. Generating missing values for simulation purposes: a multivariate amputation procedure. Journal of Statistical Computation and Simulation. 2018; 88: 2909-2930.
16. Van Buuren S, Groothuis-Oudshoorn K. Mice: Multivariate Imputation by Chained Equations in R. Journal of Statistical Software. 2011; 45.