Of mice and trees: towards proper MI using CART

Simon Fehrenbach, Tom Goldschmidt & Alice Kar

Lehrstuhl für Statistik und Ökonometrie/Universität Bamberg

Introduction

Multiple Imputation (MI) is a principled approach for analyzing incomplete data. It targets valid point estimates and standard errors by propagating uncertainty due to missingness. Proper MI requires that imputations reflect both the stochastic distribution of the missing values given the model and uncertainty in the model parameters. The mice package in R supports a wide range of imputation models, including Classification and Regression Trees (CART). Standard CART imputation introduces randomness through donor selection within terminal nodes but fits the trees in a deterministic fashion, thereby neglecting model uncertainty and potentially violating Rubin's criteria for proper MI[1].

To address this, we augmented CART imputation with a bootstrap step prior to tree estimation, while leaving the remaining MI workflow unchanged. Bootstrapping approximates draws from the observed-data posterior, inducing variability in tree structures across imputations and increasing the between-imputation variance required for proper inference. We assess this modification via different Monte Carlo simulations ($S = 500$, $n = 500$, $m = 30$ each, see our GitHub repository for detailed information), comparing its performance to standard CART and Predictive Mean Matching in terms of coverage and bias.

Algorithm

Algorithm for imputing univariate missing Data using bootstrapped CART[2]

- Draw a bootstrap sample $(\dot{y}_{obs}, \dot{X}_{obs})$ of size n_1 from (y_{obs}, X_{obs}).
- Fit $\dot{y}_{obs}$ by $\dot{X}_{obs}$ by a tree model $f(X)$.
- Predict the n_0 terminal nodes g_j from $f(X_{mis})$.
- Construct n_0 sets Z_j of all cases at node g_j, each containing d_j candidate donors.
- Draw one donor i_j from Z_j randomly for $j = 1, \ldots, n0$.
- Calculate imputations $\dot{y}_j = y_{ij}$ for $j = 1, \ldots, n0$.

The full implementation, simulation scripts, and reproducible examples are available in our GitHub repository:
`github.com/xerxim/missing_data`

Monte Carlo Simulations

Figure 1: Coverage rates and relative bias for linear parameters: $X3 = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$.
The underlying data generating processes contains only linear relationships.

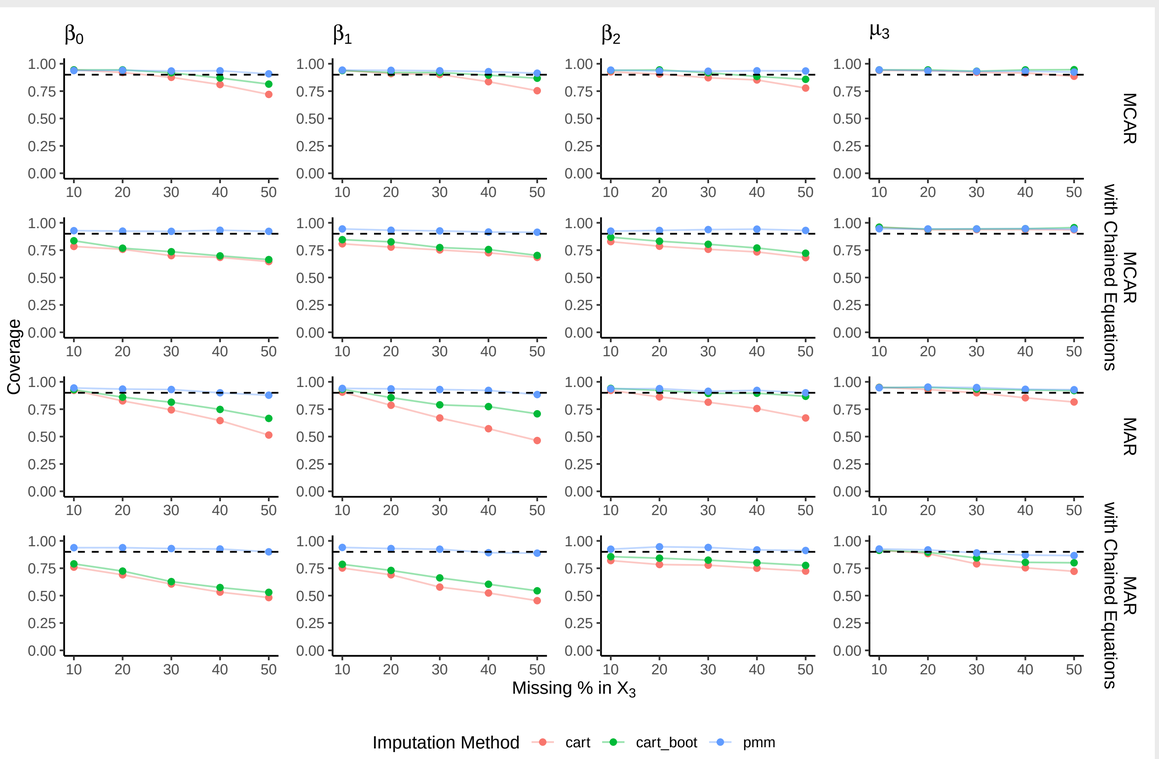
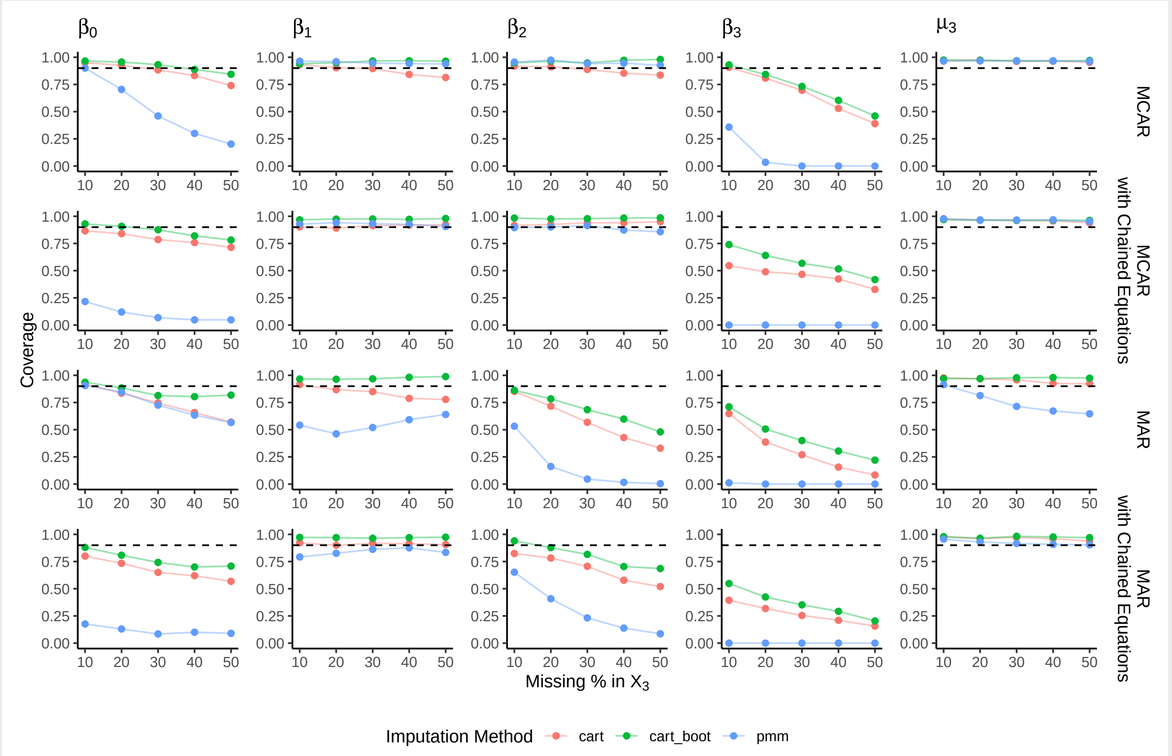
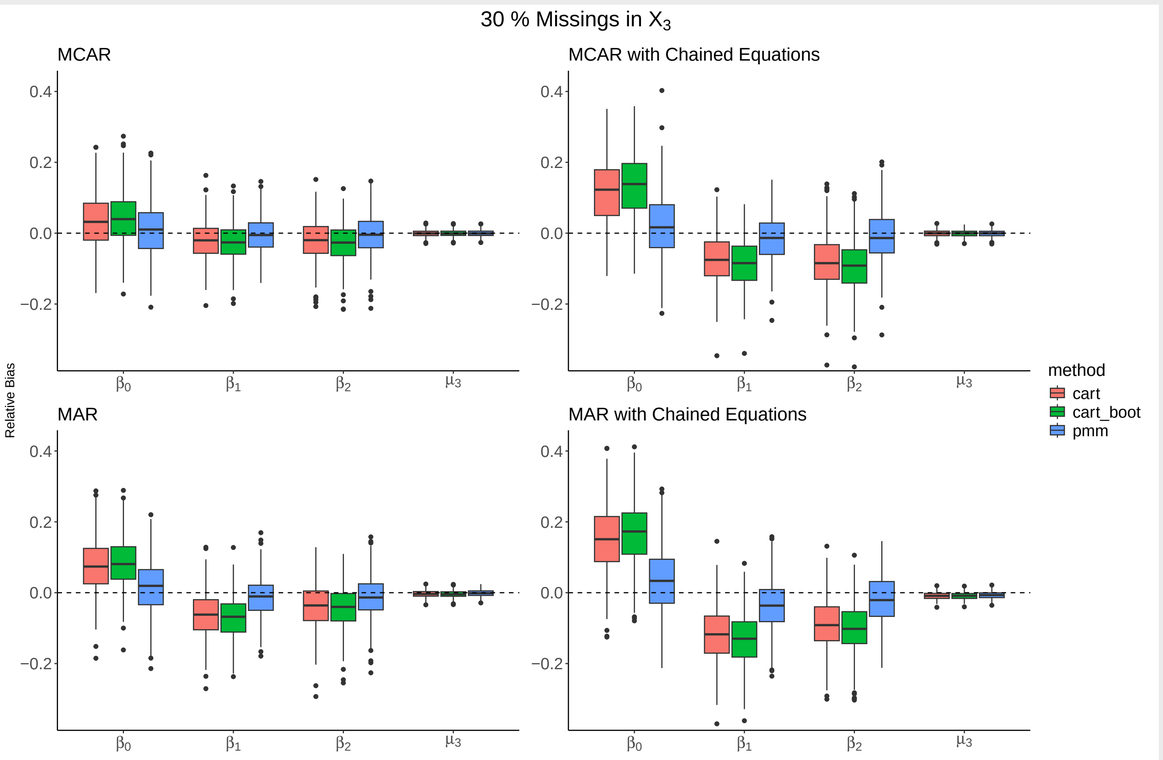
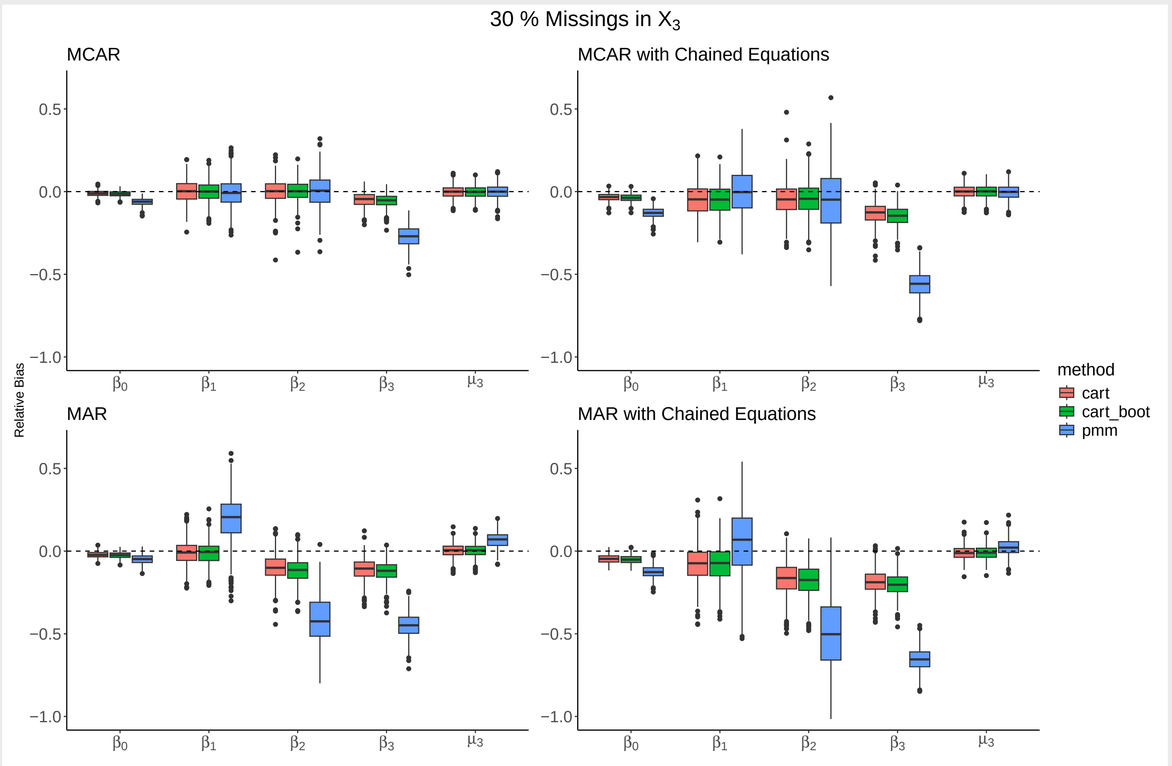


Figure 2: Coverage rates and relative bias for interaction parameters: $X3 = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \epsilon$. The underlying data generating processes contain an interaction effect.







Conclusion

The `cart_boot` method consistently yields higher coverage rates than the standard `cart` implementation, regardless of the missingness mechanism, missing rate of X_3 and the data generating process, indicating overall improved imputations. The biggest improvements were observed for data with univariate missings produced by a Missing At Random mechanism, where coverage rates were up to 25% higher than those produced by `cart`. However, `cart_boot` tends to produce slightly more biased estimates. When compared to `pmm`, `pmm` still remains the clear superior choice for purely linear data. While `cart_boot` almost equals `pmm`'s coverage rate for some cases, its bias values are noticeably larger across the board. Only when an interaction effect is introduced in the data generating process do the CART models out-perform predictive mean matching. Still, their coverage rates are low for almost every missing mechanism. Further research into how this behavior changes with weaker interaction effects, lower missing rates in X_1 and X_2 and a weaker MAR mechanism should be conducted. As such, while we do believe bootstrapping to be an overall improvement, `cart_boot` should be reserved for special cases.

References:

1. Rubin, Donald. (1987). Multiple Imputation for Nonresponse in Surveys. John Wiley & Sons, Inc.

2. van Buuren, Stef. (2018). Flexible Imputation of Missing Data. Chapter 3.5 Classification and regression trees. Available at: https://stefvanbuuren.name/fimd/sec-cart.html