Fuzzier Forests: Identifying Interactions

Ricardo A. Aguilar

under supervision of

Dr. Christina M. Ramirez

June 11st, 2021

Abstract

Fuzzy forests are a computationally feasible feature selection method for correlated, high dimensional data. This paper introduces new capabilities to fuzzy forests that allow researchers to identify interactions among features. In particular, these methods identify interactions within and across modules. Although random forest variable importance measures can account for interactions, users are unable to identify those interactions from the resulting output. Knowing which features are involved in interactions can elucidate the relationships of the features in the model. The performance of the *across* and *within* methods was evaluated using simulated data. The two methods can reliably select the interaction terms included in the true regression models. Additionally, an informal assessment for the interchangeability of correlated features provides evidence that, when using variable importance for feature selection as opposed to maximizing predictive accuracy, the correlated features do not act as proxies for one another.

Keywords: random forest, fuzzy forest, WGCNA, interactions, p >> n, feature selection, correlated data, variable importance, machine learning

1 Introduction

In an age with seemingly endless amounts of data, developing methods to analyze high-dimensional data in a computationally feasible manner has become essential to the research process. Methods that may identify important features when correlation is present, such as the least shrinkage and selection operator (LASSO, Tibshirani 1996) and smoothly clipped absolute deviation (SCAD, Fan & Li 2001), can require making parametric assumptions about the data. However, non-parametric methods are required when such assumptions cannot be made.

Random forests have widespread use and are useful for variable screening [when features are independent], but can produce biased variable importance measures (VIMs) when features are correlated (Strobl et al., 2007). Conditional VIMs (Strobl et al., 2008) can reduce this bias, but quickly become computationally infeasible as your sample size increases. Fuzzy forests (Conn et al., 2019) tackle the issue of correlated, high dimensional data in a more computationally feasible manner.

The fuzzy forest algorithm starts by partitioning data into modules in a manner determined by the user or weighted gene coexpression network analysis (WGCNA, Langfelder & Horvath 2008). The features are partitioned such that correlation of features within the same module is high, but the correlation of features in different modules is low. Features that are not correlated with any modules are placed into the *grey* module. Then, recursive feature elimination random forests (RFE-RF, Díaz-Uriarte & de Andrés 2006) are run within each module to reduce the number of features to a pre-specified proportion in that module (the selection step). All surviving features are grouped together to go through one last round of RFE-RF that results in rankings of variable importance for the selected features.

In an ideal world, one could test all possible interactions without issues and obtain empirical results that reflect the true underlying relationships between features. However, testing all possible interactions could greatly increase the runtime of the algorithm used and can introduce spurious correlations that lead to those features being mistakenly classified as

important. Examining all possible two-way interactions with p features results in

$$\binom{p}{2} = \frac{p!}{2!(p-2)!} = \frac{1}{2}(p^2 - p)$$

interactions. Clearly, the number of all possible second order interactions alone can quickly become large. Thus, the focus must be shifted to examining the subset of possible interactions that have the greatest likelihood of being important. This can be a difficult task for a researcher without any prior knowledge of interactions among the features in their data.

This paper uses simulations to investigate the hypothesis that identifying interactions can be accomplished by only examining potential interactions between the top m features ranked by variable importance. This hypothesis is based on the idea that random forests are able to account for interactions so the features ranked as the most important by RFE-RF in fuzzy forest's screening step would be the most likely to have interactions. The methods *across* and *within* were created to obtain results using two different approaches to identifying which features are most likely to be a part of an interaction.

When two features are highly correlated but only one of them is actually important, optimizing prediction accuracy might select either of the two. Fuzzy forests are an extension of random forests that select features based on variable importance rather than optimizing prediction accuracy. More specifically, fuzzy forests use random forest permutation VIMs for feature selection. An informal test that removes the top r selected features and reruns the final RFE-RF to assess the interchangeability of important features with the non-important ones was conducted.

2 Methods

2.1 *Across* and *Within* Methods

The implementation of the *across* and *within* methods occurs after the screening step of the fuzzy forest algorithm. Both methods only examine two-way interactions by default, but can examine up to three-way interactions by setting the argument `three_way` to `TRUE`. Generally, models should be hierarchically well-formulated, so the second order interactions are also examined when looking at third order interactions. The value for m, a tuning parameter specified by the user, is the number of features whose interactions will be examined. The features to be examined are chosen by their ranking based on variable importance. This rank and the resulting number of interactions examined will be determined by the method selected which will be discussed later in this section. The *within* method is restricted to interactions among features in the same module. The *across* method, however, is able to examine interactions among features in the same modules and across different modules.

The *across* method starts by ordering a data frame of all surviving features from all modules in decreasing order by variable importance. A data frame containing interactions of the specified order for the top m features is created. This data frame of interactions is then merged with the data frame of all surviving features. The final round of RFE-RF is applied to this new data frame instead of the data frame only containing the surviving features.

The *within* method begins by arranging the surviving features within their respective modules in decreasing order of variable importance. Data frames are created for each module containing interactions of the top m features within that module. These data frames of interactions are then merged with the data frame of all surviving features. Similar to the *across* method, this new data frame containing the observed data for the surviving features and the interactions goes through the final round of RFE-RF.

The value for m will greatly affect runtime, since increasing m increases the number of features examined by the final RFE-RF. Finding the top m within-module interactions when

there are k modules results in

$$\binom{m}{2} * k = \frac{m!}{2!(m-2)!} * k = \frac{k}{2}(m^2 - m)$$

total interactions. Thus, the total number of interactions examined is dependent on both m and the number of modules. Finding the top m interactions when using the *across* method introduces a total of

$$\binom{m}{2} = \frac{m!}{2!(m-2)!} = \frac{1}{2}(m^2 - m)$$

interactions. The number of interactions is no longer dependent on the number of modules since we are finding the top m overall most important features instead of the top m within each module. The interactions from the *across* method will, by design, include very similar within-module interactions as the *within* method when m is large enough.

2.2 *Remove* Test

The original paper (Conn et al., 2019) states that, in the presence of correlation, selecting features based on variable importance instead of by optimizing predictive accuracy might obtain different results. It argues that, when maximizing predictive accuracy, if two features are highly correlated then they serve as proxies for each other. This was tested by removing the top r selected features from the data frame of surviving features and rerunning the final RFE-RF. The results of the initial selection and the final selection are checked to determine if the feature of interest was in either of the two resulting data frames. The intuition behind this process is if the correlated features are interchangeable when using variable importance for feature selection and a feature that is not important was incorrectly selected in the initial results, then removing the incorrectly selected feature might increase the likelihood that the important feature it is correlated with will be selected in its place. In other words, if the proportion of times an important feature is selected increases with the *remove* test, then correlated features might still act as proxies for one another when using variable importance.

2.3 Simulations

Simulations similar to those in the original paper (Conn et al., 2019) were used to obtain results for the *across* method, *within* method, and *remove* test. Two simulations used data generated from linear models while the other three simulations used data generated from nonlinear models. The error terms are from normal distributions centered at 0 with $\sigma = 0.1$ in the linear case and $\sigma = 0.5$ in the nonlinear case. None of the true models contain an intercept term. The data for all simulations was generated using the multivariate normal distribution. The standard normal distribution was used as the marginal distribution of each X_i. The modules are made by setting the correlation of features within modules to 0.8 and features from different modules have a correlation of 0.

The first linear simulation has a total of 100 features with 25 features in each module. The first three modules, $\{X^{(1)}, ..., X^{(25)}\}$, $\{X^{(26)}, ..., X^{(50)}\}$, $\{X^{(51)}, ..., X^{(75)}\}$, contain correlated features while the last module $\{X^{(76)}, ..., X^{(100)}\}$ contains independent features. The true regression model for this simulation can be written as

$$Y_i = 5X_{i1} + 5X_{i2} + 2X_{i3} + 5X_{i76} + 5X_{i77} + 2X_{i78} + \epsilon_i$$

for $i = 1, ..., n$, where n is the total sample size. In this scenario, 100 observations were generated every simulation run. The second linear simulation has a total of 1,000 features with 100 features in each module. The first nine modules, $\{X^{(1)}, ..., X^{(100)}\}$, ..., $\{X^{(801)}, ..., X^{(900)}\}$, contain correlated features while the last module $\{X^{(901)}, ..., X^{(1000)}\}$ contains independent features. The true regression model for this scenario can be written as

$$Y_i = 5X_{i1} + 5X_{i2} + 2X_{i3} + 5X_{i901} + 5X_{i902} + 2X_{i903} + \epsilon_i$$

for $i = 1, .., n$. A total of 100 observations were generated in each simulation run for this scenario as well.

All nonlinear simulations have 100 features with the same setup for the modules, but different sample sizes and true models. Similar to the first linear simulation, all of the nonlinear simulations have $\{X^{(1)}, ..., X^{(25)}\}$, $\{X^{(26)}, ..., X^{(50)}\}$, $\{X^{(51)}, ..., X^{(75)}\}$, as the first three modules containing correlated features while the last module $\{X^{(76)}, ..., X^{(100)}\}$ contains independent features. The first simulation has

$$Y = X_1 + X_2 + 2.92X_1X_2 + \sqrt{15}X_3 + X_4^3 + X_{76} + X_{77} + 3.74X_{76}X_{77} + \sqrt{15}X_{78} + X_{79}^3,$$

for $i = 1, .., n$, as its true regression model. This simulation had 250 observations generated per simulation run. The second nonlinear simulation is almost exactly the same as the first, but it has 500 observations generated per run. The third simulation's true regression model is

$$Y = X_1 + X_2 + \sqrt{15}X_3 + X_4^3 + X_{76} + X_{77} + \sqrt{15}X_{78} + X_{79}^3 + X_3X_{78}$$

for $i = 1, .., n$. This scenario had 250 observations generated for each simulation run.

3 Results

To evaluate the performance of each method combination, the simulations were ran 100 times. The methods whose results are compared are random forest, fuzzy forest, fuzzy forest with the *remove* test, fuzzy forest with the *across* method, fuzzy forest with the *within* method, fuzzy forest with the *across* method and *remove* test, and fuzzy forest with the *within* method and *remove* test. The proportion of times the methods selected each important feature was computed. To test the performance of these methods on features that are not important, the proportion of times each method selected a particular feature that was not important was also computed.

Figure 1 shows the results for the first linear simulation. All of the fuzzy forest methods had approximately the same performance for the correlated features. For the independent

features, the fuzzy forest methods selected the important features a similar proportion of times with the exception of both of the *across* methods. These two methods struggled to identify X_{78} as an important feature. Random forest selected the important, correlated features more often than their independent counterparts. It also struggled to identify X_{78} as an important feature.

The results for the second linear simulation can be seen in Figure 2. All of the fuzzy forest methods performed very similarly for each of the specified features with only negligible differences between them. The fuzzy forest methods selected X_1 and X_2 approximately 64% of the time while selecting X_{76} and X_{77} approximately 89% of the time. Again, random forest selected the important, correlated features more often than their independent counterparts. Random forest was completely unable to identify X_{903} as an important feature.

The performance of these methods for the first nonlinear simulation can be seen in Figure 3. Again, the fuzzy forest methods performed approximately the same for each of the correlated features. The *across* methods struggled to identify X_{76} and X_{77} as important features, but the *within* methods were essentially unable to select those features. All methods selected X_{78} 100% of the time and the fuzzy forest methods selected X_{79} in over 77% of the simulation runs. Both of the *within* methods identified both of the within-module interactions in 45% to 48% of the simulation runs. Random forests were completely unable to select the features in the second interaction of the true regression model (X_{76} and X_{77}).

Figure 4 displays the results of the second nonlinear simulation. As expected, all methods performed better with more data. Fuzzy forest, fuzzy forest with the *remove* test, and both of the across methods had a notable increase in the proportion of times they were able to select X_{76} and X_{77}. The most notable changes, however, were the *within* methods' ability to select the interaction between X_{76} and X_{77} in nearly every simulation run. Although random forest was able to select most of the important features in at least 92% of simulation runs, it still struggled to identify X_{76} and X_{77} as important features.

The results of the third nonlinear simulation are shown in Figure 5. The fuzzy forest

methods obtain varying results for X_1 and X_2, but obtain similar results for X_3 and X_4. All of the methods, except for fuzzy forest and fuzzy forest with the *remove* test, were unable to select X_{76} and X_{77}. Again, X_{78} was selected in every simulation run by all of the methods. Only fuzzy forest and fuzzy forest with the *remove* test selected X_{79} 100% of the time. Both of the across methods were able to identify the across-module interaction in the majority of simulation runs.

4 Discussion

The results of the simulations suggest that all the fuzzy forest methods obtain similar results for features in modules that contain correlated features. However, the results will vary for features in modules containing independent features. Random forest's bias favoring correlated features is illustrated in all of the resulting figures. There was a notable decrease in random forest's ability to select the independent features when the number of features was significantly greater than the sample size while the fuzzy forest methods only experienced a slight decrease in performance.

The linear simulations suggest that the fuzzy forest methods introduced in this paper obtain comparable results to the original fuzzy forest algorithm when there are no interactions in the true regression model. In the presence of interactions, the *within* and *across* methods can successfully identify within-module and across-module interactions, respectively. Although the *within* method was unable to detect the importance of the independent features that are part of an interaction term, it was able to detect their interaction which implies the importance of those features. The *across* method identifies across-module interactions at the cost of introducing so many new terms that some features in the true model might not be selected. All of the methods with the *remove* test have very similar results compared to their counterparts. If correlated features were still interchangeable when using variable importance, then the *remove* tests would consistently yield better results than their coun-

terparts. This provides evidence that correlated features are not proxies for each other when using variable importance for feature selection.

5 Conclusion

The era of big data introduces many new challenges to researchers, especially when features are highly correlated. Fuzzy forests offer a solution that is computationally feasible and produce less biased results. This paper introduced two new methods of identifying interactions while using the fuzzy forest algorithm. Both of the new methods, *across* and *within*, were able to successfully select the interaction terms in the true regression models while obtaining comparable results in the absence of interactions.

Even though the methods introduced in this paper work as intended, their performance was tested in only a few of the many possible situations that can occur with real-world data. Although, both methods can include three-way interactions, no simulations were ran to test the methods' performance on third order interactions. Additionally, none of the simulations contained across-module interactions when both modules have correlated features. Another limitation is that fuzzy forests only use random forest permutation VIMs to rank features. There is evidence that Gini importance can be more successful than permutation importance at capturing interactions (Wright et al., 2016). Thus, other VIMs should be looked into as well. Furthermore, the across method currently uses the raw value of variable importance, so the results might be improved if those values were scaled in some manner to make them more comparable. Perhaps developing a rule of thumb for the tuning parameter m at various ratios of sample size to number of features would also be helpful. Improving these results might help researchers overcome the challenges associated with correlated, high-dimensional data.

References

Conn, D., Ngun, T., Li, G., & Ramirez, C. M. (2019). Fuzzy forests: Extending random forest feature selection for correlated, high-dimensional data. *Journal of Statistical Software, Articles*, *91*(9), 1–25.
URL https://www.jstatsoft.org/v091/i09

Díaz-Uriarte, R., & de Andrés, S. A. (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, *7*(1), 3.

Fan, J., & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, *96*(456), 1348–1360.

Langfelder, P., & Horvath, S. (2008). WGCNA: an r package for weighted correlation network analysis. *BMC Bioinformatics*, *9*(1).

Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., & Zeileis, A. (2008). Conditional variable importance for random forests. *BMC Bioinformatics*, *9*(1), 307.
URL https://doi.org/10.1186/1471-2105-9-307

Strobl, C., Boulesteix, A.-L., Zeileis, A., & Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, *8*(1), 25.
URL https://doi.org/10.1186/1471-2105-8-25

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

Wright, M. N., Ziegler, A., & König, I. R. (2016). Do little interactions get lost in dark random forests? *BMC Bioinformatics*, *17*(1).

A Figures

Figure 1: Results for the simulation with a linear true regression model where $n = 100$, $p = 100$, and $Y = 5X_1 + 5X_2 + 2X_3 + 5X_{76} + 5X_{77} + 2X_{78}$

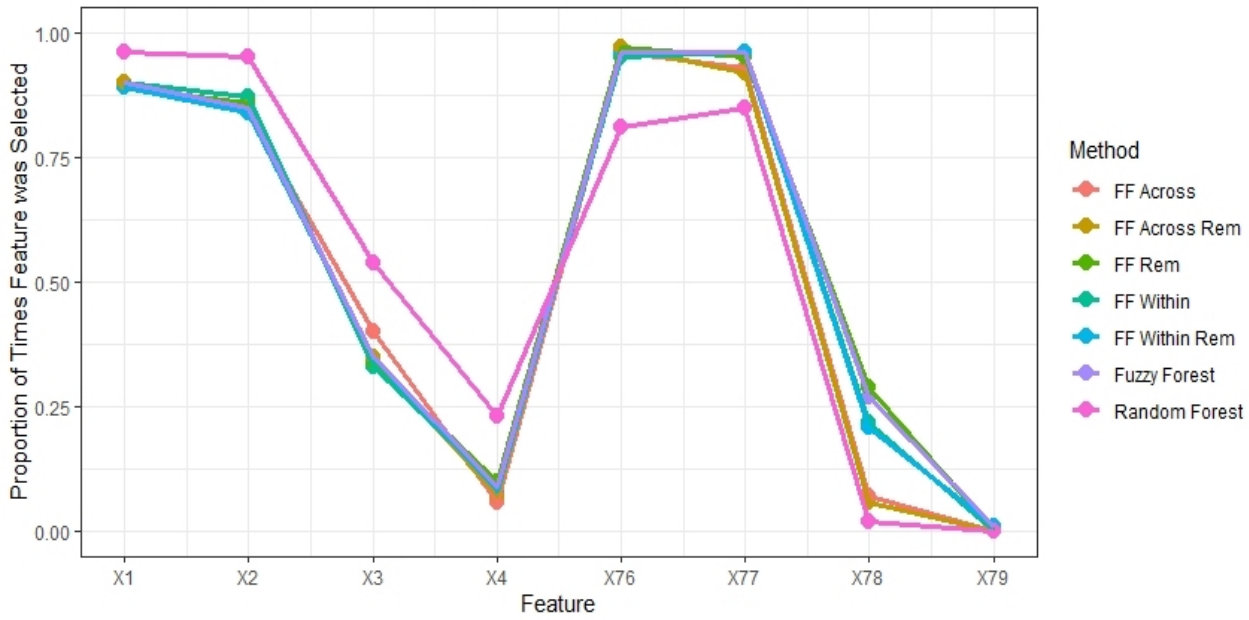


Figure 2: Results for the simulation with a linear true regression model where $n = 100$, $p = 1000$, and $Y = 5X_1 + 5X_2 + 2X_3 + 5X_{901} + 5X_{902} + 2X_{903}$

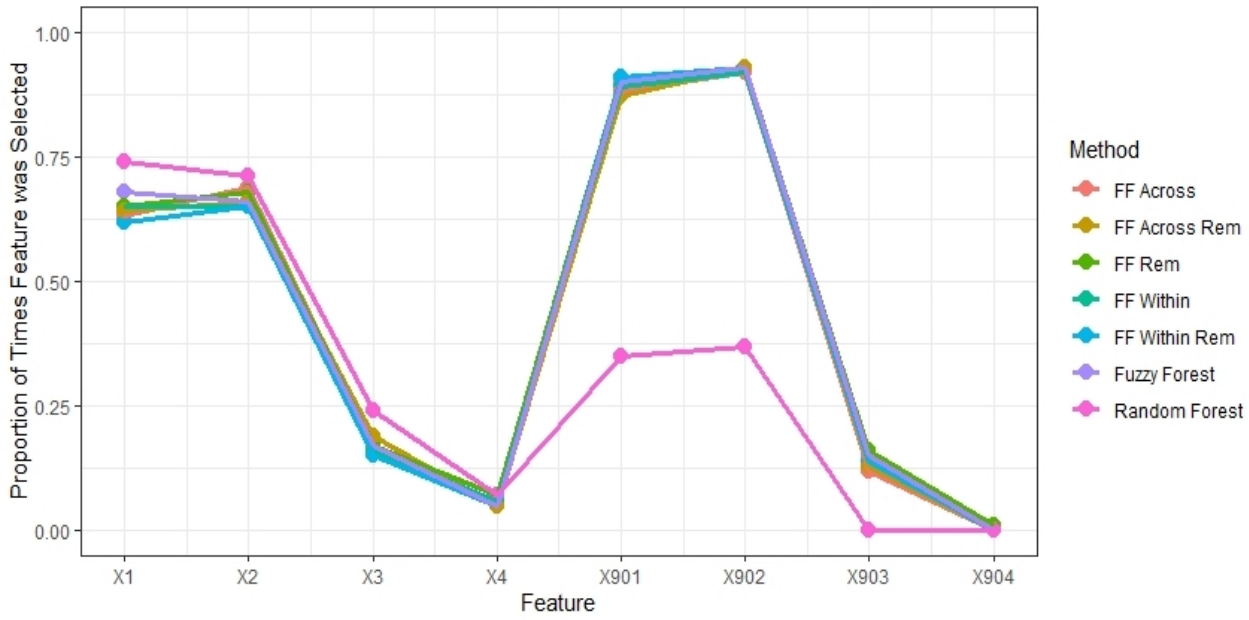


Figure 3: Results for the simulation with a nonlinear true regression model where $n = 250$, $p = 100$, and
$Y = X_1 + X_2 + 2.92X_1X_2 + \sqrt{15}X_3 + X_4^3 + X_{76} + X_{77} + 3.74X_{76}X_{77} + \sqrt{15}X_{78} + X_{79}^3$

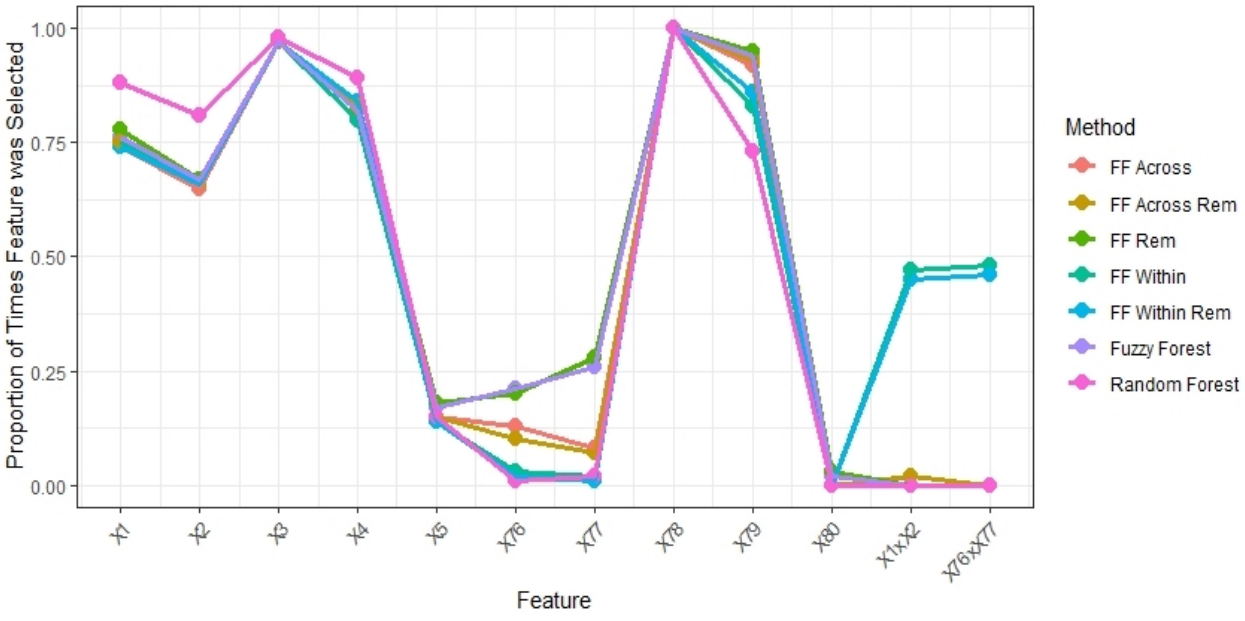


Figure 4: Results for the simulation with a nonlinear true regression model where $n = 500$, $p = 100$, and
$Y = X_1 + X_2 + 2.92X_1X_2 + \sqrt{15}X_3 + X_4^3 + X_{76} + X_{77} + 3.74X_{76}X_{77} + \sqrt{15}X_{78} + X_{79}^3$

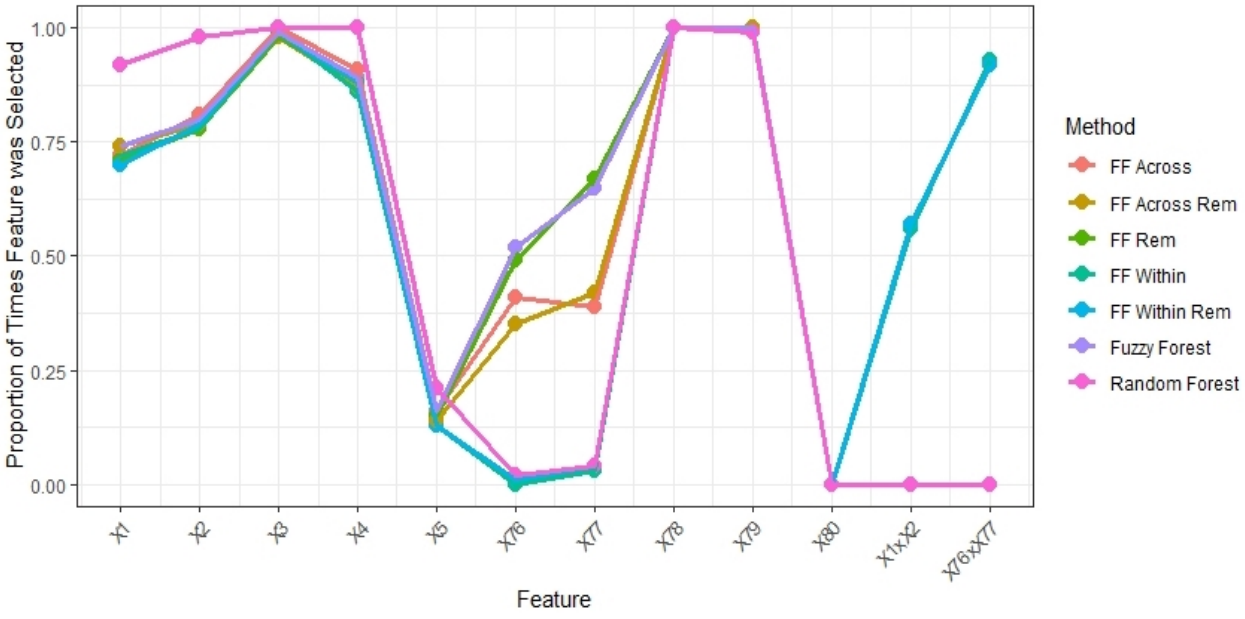
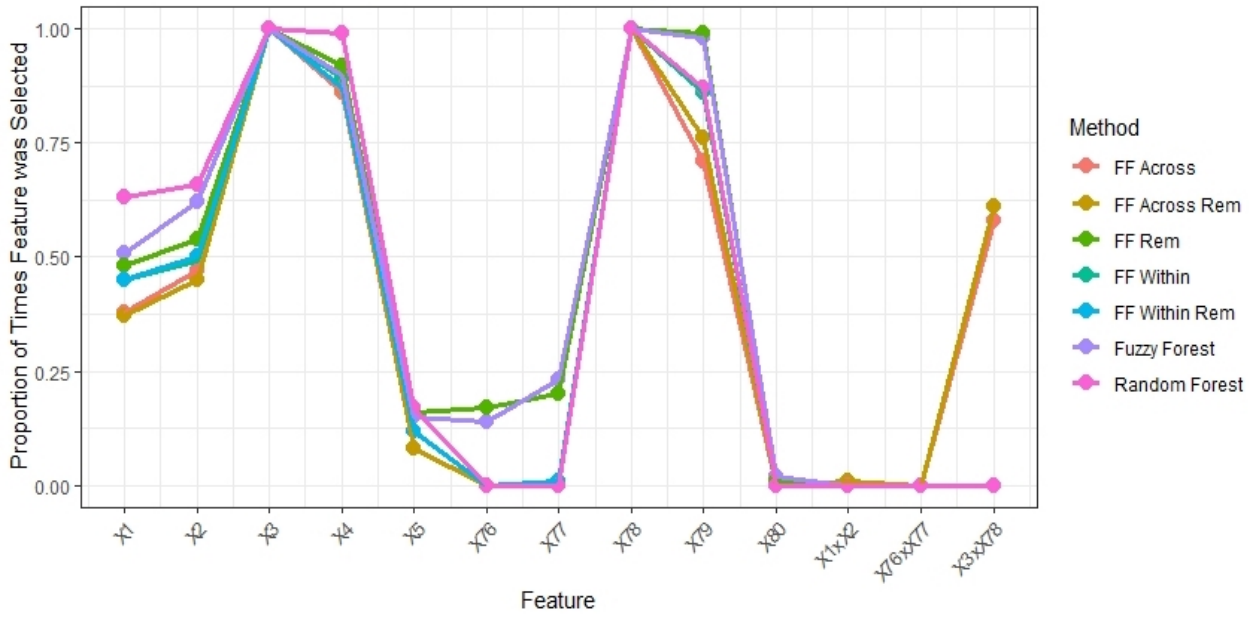


Figure 5: Results for the simulation with a nonlinear true regression model where $n = 250$, $p = 100$, and $Y = X_1 + X_2 + \sqrt{15}X_3 + X_4^3 + X_{76} + X_{77} + \sqrt{15}X_{78} + X_{79}^3 + X_3X_{78}$



B Supplementary Materials

All code and supplementary materials can be found in the GitHub repository:

https://github.com/raguilar2/fuzzier_forest

Affiliation:

Ricardo Aguilar
Department of Biostatistics
UCLA Fielding School of Public Health
E-mail: raguilar2@ucla.edu