FREEtree: A Tree-based Approach for High Dimensional Longitudinal Data With Correlated Features

Yuancheng Xu[1], Athanasse Zafirov[2], R. Michael Alvarez[3], Dan Kojis[4], Min Tan[5], and Christina M. Ramirez[6]

[1]*Department of Mathematics, Southern University of Science and Technology*
[2]*Anderson School of Management, University of California, Los Angeles*
[3]*Division of Humanities and Social Sciences, California Institute of Technology*
[4]*Department of Statistics, The University of Wisconsin–Madison*
[5]*Department of Mathematics, Sichuan University*
[6]*Department of Biostatistics, Fielding UCLA School of Public Health*

Abstract

This paper proposes FREEtree, a tree-based method for high dimensional longitudinal data with correlated features. Popular machine learning approaches, like Random Forests, commonly used for variable selection do not perform well when applied to high dimensional longitudinal data with correlated features. FREEtree deals with longitudinal data by using a piecewise random effects model. It also exploits the network structure of the features by first clustering them using Weighted Gene Coexpression Network Analysis (WGCNA). It then conducts a screening step within each cluster of features and a selection step among the surviving features, which provides a relatively unbiased way to select features. By using dominant principle components as regression variables at each leaf and the original features as splitting variables at splitting nodes, FREEtree maintains its interpretability and improves its computational efficiency. The simulation results show that FREEtree outperforms other tree-based methods in terms of prediction accuracy, feature selection accuracy, as well as the ability to recover the underlying correct structure.

Keywords: longitudinal data, random effects, regression trees, variable selection, machine learning interpretability.

1 Introduction

Longitudinal or clustered data, where observations within a unit (cluster) are more correlated than observations from other units (clusters), are very common in areas such as social science and medical research. Further, the data may contain a large number of correlated features relative to the number of observations (high dimensional data). The goal of this paper is to extend tree-based algorithms to high dimensional longitudinal data with correlated features and to develop a relatively interpretable data mining technique for feature selection and prediction.

Tree based algorithms began to gain momentum with the appearance of CART (classification and regression trees) algorithm (Breiman et al., 1984) [1]. They are widely used in statistical machine learning due to their interpretability, relatively high computational efficiency, and their nonparametric and nonlinear nature. Briefly, a binary decision tree based algorithm recursively partitions the parameter space until it meets a stopping criteria, and uses a piecewise model on each subset of the data. The algorithm is greedy and does not take into account any correlation or longitudinal structure.

Segal[13] made the first attempt to deal with longitudinal data by using regression trees, and proposing a new split function depending on the covariance structure of multiple responses. However, this method cannot deal with time-varying covariates (only the responses, instead of covariates, vary with time in his setting) and all the observations within a unit end up in one terminal node. Mixed-effects longitudinal trees (MELT)(Cho et al., 2014)[4] fully explore the shape of the data with respect to time by fitting low degree polynomials and splits on the coefficients. The objective of MELT is to identify different shapes of time among units. However, MELT only deals with time-invariant covariates and it is not optimized for prediction.

Sela and Simonoff (2012) [14] proposed the RE-EM tree, which uses a random effects model to deal with longitudinal structure, where the fixed effect is modelled as a standard regression tree CART. The random effects and fixed effect are estimated alternatively, which is similar to the EM algorithm. Later, a new version of RE-EM tree was proposed by Simonoff and Fu (2015)[6] where the implementation of the fixed effect was replaced by the conditional inference trees of Hothorn et al. (2006)[8] to reduce bias. RE-EM tree can deal with time-varying covariates and observations within a unit can end up in different terminal nodes.

The generalized linear mixed-effects model trees (GLMM tree) algorithm (M. Fokkema et al., 2017)[5] adopts a more general approach than the RE-EM tree. The GLMM tree also uses a random effect model, but with the fixed effect modelled as a piecewise generalized linear model, i.e. a regression model tree with a generalized linear model, instead of a constant, at each leaf. The local fixed-effects regression model at every terminal node and the global random effects, which are common to all the observations within a unit, are estimated in a similar fashion as in RE-EM tree. The GLMM tree provides more flexibility in the model of fixed effect and can be used to detect treatment effects (see 4.1). The GLMM tree approach will be discussed in more detail in 2.2.

It is known that Random Forests variable selection is biased when there is correlation. Fuzzy Forests (Conn, Ramirez et al., 2015)[2] was developed to address correlation within the predictors in the setting where $p >> n$. The first step is to explicitly cluster features using weighted correlation networks [16](reviewed in 2.1). Then a feature screening step is conducted within each cluster using Recursive Feature Elimination Random Forests (RFE-RFs) [3]. Finally a feature selection step is done within the features selected from the screening step, allowing clusters to interact with each other. The screening step and the selecting step enables Fuzzy Forests to select features in a relatively unbiased way in the presence of highly correlated features. The Fuzzy Forests methodology has been used in a number of applied research articles, for example [12, 9].

This paper proposes the Fuzzy Random Effect Estimation tree (FREE-

tree), which takes advantage of the powerful feature selection approach of Fuzzy Forests, as well as the flexible framework of the GLMM tree to deal with the longitudinal structure of the data.

The remainder of the article is organized as follows: Section 2 reviews the building blocks of FREEtree before section 3 explains FREEtree algorithms in detail. Section 4 provides simulation results of FREEtree on two simulated data sets, one with time-treatment interaction and one without it. Section 7 discusses some of the future work and the last section concludes the paper.

2 A review of WGCNA and GLMM tree

2.1 WGCNA

Weighted correlation networks (WGCNA) have been used in many applications to examine the network structure of covariates [10, 11, 7]. This is an unsupervised learning method. In order to construct the network, WGCNA does the following: (1) choose a similarity function for feature X^u and X^v, denoted by s_{uv}. A common choice is $Corr(X^u, X^v)$ where Corr is the Pearson correlation. Then compute the similarity matrix $S = [s_{uv}]$. (2) Transform the similarity matrix X by the adjacency matrix $A = [a_{uv}]$ where $a_{uv} = s_{uv}^{\beta}$ which results in a soft-thresholding network. The β is chosen according to the scale-free criterion [16].(3) Convert adjacency matrix A to the topological overlap matrix (TOM) W through Eq.(1) where $q_{uv} = \sum_{r=1}^{p} a_{ur}a_{rv}$ and $c_u = \sum_{r=1}^{p} a_{ur}$. (4) Use a hierarchical clustering tree algorithm to find clusters using TOM. The reason that hierarchical clustering algorithm uses TOM instead of the adjacency matrix A is that using TOM may lead to more distinct modules [16].

$$w_{uv} = \frac{q_{uv} + a_{uv}}{\min\{c_u, c_v\} + 1 - a_{uv}} \tag{1}$$

Weighted correlation network analysis (WGCNA)[16] can be used for clustering covariates where covariates within each module are highly correlated and genes from different modules are approximately uncorrelated. Covariates that are not assigned to any clusters are placed in the grey module. That is, each grey covariate in the grey module is roughly uncorrelated to any other covariates and can be viewed as a cluster on its own. Note that in the context of machine learning, we can view each feature as a gene (to keep consistent with the genetics literature) and therefore WGCNA can identify modules of highly correlated features.

2.2 GLMM tree

The rational behind the Generalized Linear Mixed-Effects Model tree (GLMM tree) [5] is that a global generalized linear mixed-effect model may not fit the data well. However, if additional splitting variables are available, we can fit the data with piece-wise models by partitioning the data with these splitting variables.

To fix ideas, suppose that in our dataset the t^{th} observation of cluster i consists of covariates x_{it} and response y_{it}. For example, cluster i may stand for the i$^{\text{th}}$ patient and t, the time of the measurement. Then a global Generalized

Linear Mixed-Effects model (GLMM) is given by

$$E[y_{it}|x_{it}] = \mu_{it}; \quad g(\mu_{it}) = x_{it}^T\beta + z_i^T b_i; \tag{2}$$

where g is the link function and β is a vector of fixed-effect regression coefficients (as opposed to the power function described in WGCNA). For a mixed-effect model with only random intercept, z_i is just constant 1 and b_i is the random intercept associated with cluster i. When random slopes are involved, z_i is the design vector which is a subset of x_{it} and b_i is the random vector with each component corresponding to the random deviation of the slope from the fixed-effect. For simplicity, we assume that the link function g is the identity function and the mixed-effects model with only random intercept is adopted. That is, we are using a linear mixed-effect model with only a random intercept from now on, as the following:

$$\mu_{it} = x_{it}^T\beta + b_i \tag{3}$$

In many cases, the Linear Mixed-Effect model (LMM) in Eq(3) may not fit the data well because the assumption that the underlying fixed-effect model is a linear function is too restrictive. It makes more sense to approximate the fixed-effect structure with a piece-wise linear model instead of a global linear model. GLMM tree uses a model-based recursive partitioning (MOB) algorithm [15] which partitions the dataset using splitting variables and find better-fitting local LMM models. MOB iterates the following: fit a parametric model (such as LMM) to the dataset and then adopts parameter stability tests on each of splitting variables by computing a p-value for every splitting variable. If the smallest p-value is below the significant level α, the dataset is split into two subsets using the splitting variable value with the smallest p-value, with the split point for that variable chosen to minimize the instability. Therefore only significant splitting variables will be used for splitting at the node of a GLMM tree. More details of the parameter stability test are described by Zeileis[15]. The resulting GLMM tree has the following form,

$$\mu_{it} = x_{it}^T\beta_{j(it)} + b_i \tag{4}$$

where $j(it)$ is the index of terminal node that t^{th} observation of cluster i belongs to. Note that the fixed-effect is now a piece-wise linear function of covariates and the random intercept is global in the sense that it only depends on the cluster, instead of the terminal node. The GLMM tree is trained by iteratively estimating the fixed-effect (a linear mixed-effects tree) assuming random effects are known and estimating random effects by assuming fixed effects are known until convergence.

The R package, **glmertree**[5] implements the GLMM tree. In the following section, we use LMM tree for simplicity. That is, we assume the link function g is the identity function. The function `lmertree()` is used in this package.

3 The FREEtree estimation method

The goal of FREEtree is to select features and then use the chosen features to make predictions. The advantage of having fewer features is parsimony and increased interpretability. At the heart of the algorithm lies a binary decision tree splitting strategy that is easily interpretable. While CART and many other

methods are usually biased towards selecting correlated features while ignoring independent ones in feature selection, FREEtree reduces this bias by clustering features by their correlation pattern and screening features within each cluster, while letting them interact. The resulting features are used to fit a LMM tree, which includes a linear regression model at the end of each leaf that also considers a random effect at the patient level. The predictive power mostly comes from LMM tree, which fits the data with piecewise linear function of covariates plus a random effect, instead of a piecewise constant function like CART and RE-EM tree. However, in order to regress on covariates, feature selection is necessary because linear regression requires that the sample size be sufficiently larger than the number of parameters for identifiability. FREEtree integrates feature selection and prediction in a natural way and is particularly useful when p is larger than n.

3.1 Notation

The training dataset consists of patients $i = 1, 2, ..., n$, who are measured at time $t = 1, 2, .., T$. To simplify the notation, we assume balanced data here, though this is not required for the FREEtree algorithm. Each patient has three types of features as followings:

- `var_select` X: Features of length p that will be chosen from.
- `fixed_regress` R: Features that will be used for regression in every tree. In longitudinal settings, this could be time or higher order of time.
- `fixed_split` S: Features that will be considered as splitting variables in every tree.

The value of features of patient i of time t is denoted by x_{it}, r_{it} and s_{it} respectively. Note that `var_select`, `fixed_regress` and `fixed_split` can be empty. The user can use their own prior knowledge to determine which type a certain feature belongs to. The goal of FREEtree is to select important features from `var_select` and use the selected features as well as `fixed_regress` and `fixed_split` to give the final prediction.

3.2 The FREEtree algorithm

The FREEtree algorithm consists of a feature selection step and a prediction step. First assume that `fixed_regress` is not empty. The case where it is empty will be discussed in section 3.4.

The feature selection step has three steps: clustering, screening, and selection. During the clustering step, features in `var_select` are clustered by WGCNA into modules, which includes a grey module and non-grey modules. The gray module includes all covariates that have low connectivity and can be viewed as roughly independent. Features in the same non-grey module are highly correlated/connected with each other and have lower correlation or connectivity with the features from other modules. Let there be m modules selected by WGCNA. Denote the modules of `var_select` by $\{P_1, ..., P_m\}$ and let $p_l = |P_l|$ so that $\sum_{l=1}^{m} p_l = p$. Without loss of generality, denote the last module P_m as the grey module.

For the screening step, features are selected within each module as following: For module l ($l = 1, 2, ..., m$), use `fixed_regress` as regression variables and use P_l as well as consider `fixed_split` for splitting variables to fit a LMM tree. The selected features from module l are the set of features P_l^S used in the LMM tree that are not included in `fixed_split`. The result of the screening step is a set of screened features $\{P_1^S, ..., P_m^S\}$.

The final selection step allows the selected features from each modules to interact with each other. FREEtree uses all of the screened features $\{P_1^S, ..., P_m^S\}$ from the screening step and treats `fixed_split` as splitting variables, then uses `fixed_regress` as regression variables to fit a LMM tree. The final selected features from `var_select` are the features used by this LMM tree that are not included in `fixed_split`, denoted by x^S.

Finally, at the prediction step, a LMM tree is fitted using `fixed_split` and X^s as splitting variables and using `fixed_regress` and X^s as regression variables. The prediction is provided by this final LMM tree. Note that the final selected features X^s from `var_select` are used both as splitting and regression variables, which fits the data in a more flexible way than just regressing on `fixed_regress`.

3.3 Another strategy for feature selection

The screening and selection steps help reduce bias in feature selection by eliminating features in correlated modules and thus protecting independent features from being ignored by LMM tree. However, if the number of non-grey modules is large and there are many correlated features after screening step, the independent features are still in the danger of being ignored at the selection step. In order to fully protect independent features, another strategy of feature selection is proposed, which is particularly helpful if the number of correlated feature is large compared with independent features. Users can set Fuzzy=False to use this strategy. If Fuzzy=True, the strategy in section 3.2 will be adopted.

At the screening step, features within each non-grey modules $\{P_1, ..., P_{m-1}\}$ are screened into $\{P_1^S, ..., P_{m-1}^S\}$. That is, use P_l ($l = 1, 2, ..., m-1$) and `fixed_split` as splitting variables and use `fixed_regress` as regression variables to fit a LMM tree and choose features P_l^S used by the tree and not contained in `fixed_split`. Note that for now we don't screen within the grey module P_m. Then we select features within the screened features $\{P_1^S, ..., P_{m-1}^S\}$ from the non-grey groups by using all of the screened features and `fixed_split` as splitting variables, and `fixed_regress` as regression variables to fit a LMM tree. The selection step allows the non-grey modules to interact with each other producting $\{Q_1^S, ..., Q_{m-1}^S\}$ with $Q_l^S \subset P_l^S$ for $l = 1, 2, ..., m-1$. Then we fit a LMM tree using `fixed_split` and features in the grey module as splitting variables, and regress on `fixed_regress` as well as $\{Q_l^S\}_{l=1}^{m-1}$. The set of selected features from the grey module are the ones used in this LMM tree that are not included in `fixed_split`, which is denoted by Q_m^S. The final result of feature selection is $\{Q_l^S\}_{l=1}^m$, denoted by X^s. A final LMM tree for prediction is fitted using `fixed_split` and X^s as splitting variables and using `fixed_regress` and X^s as regression variables.

3.4 Use principal components in the absence of regressors

Suppose that we do not have a natural choice for `fixed_regress` and set it to empty. One obvious way to do feature selection and prediction is to use RE-EM tree [14] with an averaged value at each leaf instead of a linear regression model. The disadvantage is that the assumption of the underlying true model being a RE-EM tree, a piecewise constant function plus random intercept, which can be too restrictive.

It is more flexible to fit the underlying model with a piecewise linear function in addition to random intercept. Therefore another method that is proposed which could have more power in feature selection and predictionis to use the dominant principal components (PC) of the non-grey modules as intermediate regressors. The idea behind it is that in linear regression, using the dominant principle components as regressors has a comparable power in terms of prediction as using all the covariates as regressors, although interpretability is lost. However, FREEtree, even if it uses PCs, is still interpretable because PCs are used only in the step of feature selection and the selected features are determined by the non-terminal nodes of the tree, instead of PCs or any other regressors. The first PCs of non-grey modules are used for simplicity, though more dominant features can be used. Note that we do not use PCs of grey module since features within grey module are roughly independent and thus it is likely that there may be no dominant PCs.

For the screening step, features from non-grey modules P_l (l=1,2,..,m-1) are selected by fitting a LMM tree using the first PC of P_l as regression variables and use P_l and `fixed_split` as splitting variables. If Fuzzy=True, for the grey module P_m, a RE-EM tree is fitted using `fixed_split` and the features used in the node of RE-EM tree are selected. Denote the screened features by $\{P_l^S\}_{l=1}^m$. For the selection step, final features X^S are obtained by selecting from the screened features. That is, fit a RE-EM tree using $\{P_l^S\}_{l=1}^m$ and select those appeared in the nodes of the RE-EM tree. In the prediction step, a LMM tree is fitted using X^S and `fixed_split` as splitting variables and X^S as regression variables.

If Fuzzy=False, final non-grey features $\{Q_l^S\}_{l=1}^{m-1}$ are obtained by selecting from screened features $\{P_l^S\}_{l=1}^{m-1}$ from non-grey modules. That is, use all the $\{P_l^S\}_{l=1}^{m-1}$ as splitting variables to fit a RE-EM tree and select features used in the node of RE-EM tree and not contained in `fixed_split`. Then the selected grey-features Q_m^S are obtained by fitting a LMM tree using the grey module P_m and `fixed_split` as splitting variables and $\{Q_l^S\}_{l=1}^{m-1}$ as regression variables. The final set of selected features X^S is $\{Q_l^S\}_{l=1}^m$. The prediction is given by a LMM tree using X^S and `fixed_split` as splitting variables and using X^S as regression variables.

4 Simulation

4.1 Design of simulations

We provide simulations to examine the utility of FREEtree in terms of feature selection, prediction and estimating the underlying model structure. In all simulations, the training dataset has n patients (we will allow n to vary) and each patient has $p = 400$ features X to be selected along with `fixed_split` and

`fixed_regress`. The features, X, are grouped into 4 modules $\{X^{(1)}, ..., X^{(100)}\}$, $\{X^{(101)}, ..., X^{(200)}\}$,$\{X^{(201)}, ..., X^{(300)}\}$ as well as $\{X^{(301)}, ..., X^{(400)}\}$. Each feature $X^{(i)}$ is generated from a multinomial normal distribution with mean 0 and variance 1. The features from different modules are uncorrelated and features within the first three modules are correlated with correlation 0.8, while features within the last module are uncorrelated. Therefore, the first three modules are called non-grey modules and the final module is the grey module, according to the conventions in WGCNA.

The first simulation includes a time by treatment interaction where different treatments corresponds to different patterns of response with respect to time. For simplicity, we assume two treatments here, treatment1 and treatment2. The true model for patient i at time t is given by

$$y_{it} = f(X_{it}) + (t-3)^2 \mathbb{1}_{treatment1} - (t-3)^2 \mathbb{1}_{treatment2} + b_i + \epsilon_{it}$$

where $\mathbb{1}$ is the indicator function, ϵ_{it} is the error is drawn from normal distribution and f is given by

$$f(X) = 5X^{(1)} + 2X^{(2)} + 2X^{(3)} + 5X^{(2)}X^{(3)} + 5X^{(301)} + 2X^{(302)} + 2X^{(303)} + 5X^{(302)}X^{(303)}$$

Here only 6 variables out of 300 are important. The other variables are noise. We use *treatment* as `fixed_split` and use $time(t)$ and $time2$ (t^2) as `fixed_regress`. Here `var_select` is X with important features being $X^{(1)}, X^{(2)}$, $X^{(3)}, X^{(301)}, X^{(302)}$ and $X^{(303)}$. Since we have a natural choice for `fixed_regress`, in *time* and *time*2, we adopt the method described in section 3.2 and section 3.3.

In a second simulation, we consider a mixed effects model given by

$$y_{it} = f(X_{it}) + b_i + \epsilon_{it}$$

where f and ϵ_{it} are the same as in the first simulation and b_i is the random intercept corresponding to patient i which is drawn from normal distribution with mean 0 and variance 3. Random intercepts of different patients are independent. Since now we do not have a natural choice for `fixed_regress`, we adopt the method described in section 3.4. That is, during the screening step, we regress on the first principal components of non-grey modules to select features from non-grey modules.

In both simulations, a validation set of 100 patients is used for tuning parameters and a test set of 100 patients is used for measuring root mean squared error on future observations. The prediction does not include random intercepts because they cannot be estimated from unknown patients. The performances of Random Forests and Fuzzy Forests in the following sections are measured by running 50 times using different random seeds.

4.2 Predictive performance

In this section, we first consider the dataset with time-treatment interaction detailed in the previous section. We compare the predictive performance of FREEtree, Random Forests, Fuzzy Forests and LMM tree. For Random Forests and Fuzzy Forests, `var_select` $\{X^{(v)}\}_{v=1}^{400}$, `fixed_regress` *time* and *time*2 and `fixed_split` *treatment* are used as covariates. *Time*, *time*2 and *treatment* are

manually put into the "grey" module in Fuzzy Forests because the time variables are uncorrelated with $\{X^{(v)}\}_{v=1}^{400}$ in the generating process and treatment is categorical which WGCNA cannot deal with directly. For LMM tree, treatment and $\{X^{(v)}\}_{v=1}^{400}$ are specified as splitting variable and *time*, *time*2 are used as regression variables. That is, unlike FREEtree, there is no feature selection before using LMM tree. Note that in LMM tree we do not regress on $\{X^{(v)}\}_{v=1}^{400}$ because linear regression requires that the sample size be greater than the number of parameters in linear regression model. Fig.1 shows the results on this dataset. FREEtree outperforms other methods when the sample size is relatively large. When the sample size is relatively small, FREEtree does not have an strong advantage since it has a linear regression model at each leaf and thus has a lot more parameters to estimate necessitating a larger sample size.

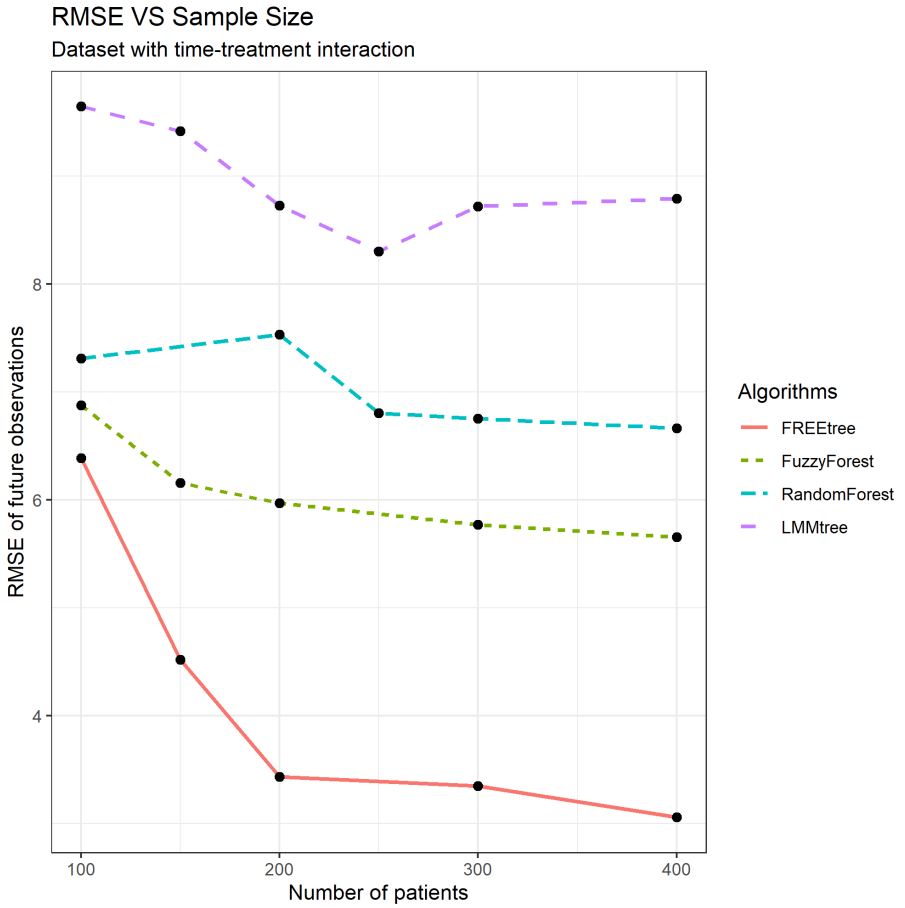

Fig. 1: Predictive performance using the time-treatment interaction dataset

Fig.2 gives the results of the performance on the simulated dataset with only random intercepts, a special case of longitudinal structure. The RMSE of Random Forests, Fuzzy Forests, RE-EM tree and FREEtree are given. Only $\{X^{(v)}\}_{v=1}^{400}$ are used in these algorithms. It shows that FREEtree has better predictive performance than other algorithms and performs better when the

sample size is larger. Note, that unlike the case in the previous simulation, FREEtree does well even when n is relatively small because the dataset structure here is much simpler.

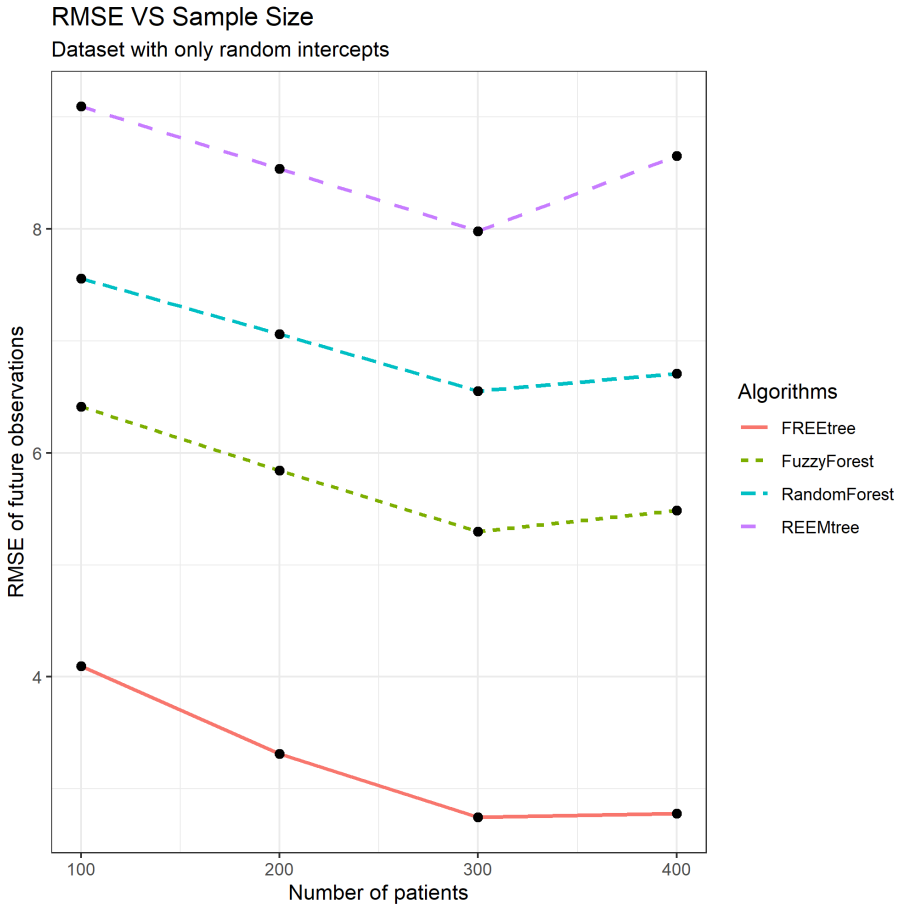

Fig. 2: Predictive performance on the dataset with only random intercepts

4.3 Feature selection performance

In this section, we compare the performance of feature selection from FREEtree and Fuzzy Forests, which is designed for feature selection. For Fuzzy Forests, we computed the proportion of times each feature was selected as important over 50 simulation runs on the same training set with different seeds and/or tuning parameters. In each run, the top 12 features are selected in the first simulation with time-treatment interaction dataset and top 10 features are chosen in the second simulation using dataset with only random intercepts. For FREEtree, the final chosen features are presented.

In the first simulation, with the true features being $X^{(1)}, X^{(2)}, X^{(3)}, X^{(301)}, X^{(302)}$, $X^{(303)}$, $treatment$, $time$ and $time2$, Fuzzy Forests successfully identified $X^{(1)}, X^{(2)}$, $X^{(3)}, X^{(301)}, X^{(302)}, X^{(303)}$ with probability 1 but missed $time$ and $time2$ com-

pletely (selected 0 times) regardless of the overall sample size. It identifies *treatment* with probability 1 when $n \geq 150$. The results of Fuzzy Forests are shown in Fig.4. As for FREEtree, since treatment, time and time2 are explicitly specified to use as splitting and regression variable respectively, we only need to examine the final selected features from `var_select` $\{X^{(v)}\}_{v=1}^{400}$. Fig.3 gives results for this simulation and it shows that in this dataset FREEtree can recover the true important features when $n \geq 150$.

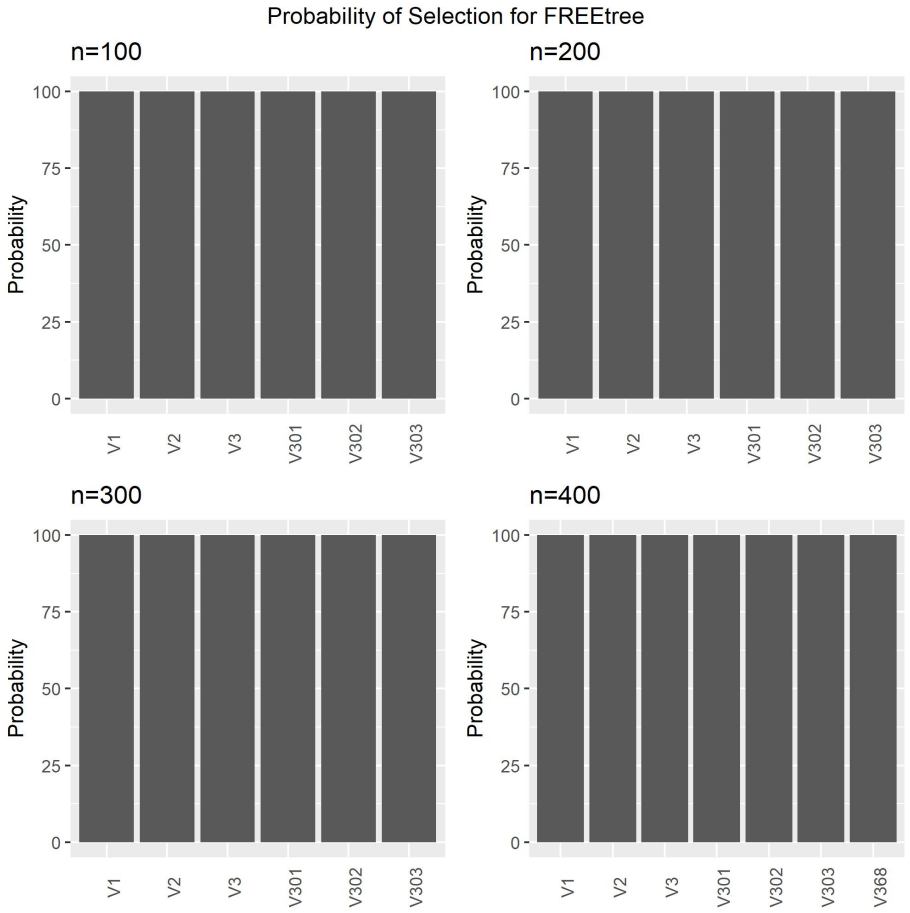

Fig. 3: The selected feature of FREEtree with different sample size n on the dataset with time-treatment interaction.

In the second simulation where the true generating process only includes random intercepts, the feature selection performance of Fuzzy Forests and FREEtree were also studied. Fig.6 shows the results of Fuzzy Forests, which recovers all the important features correctly. Fig.5 shows that FREEtree can also recover all the important features for all of the sample sizes tested.

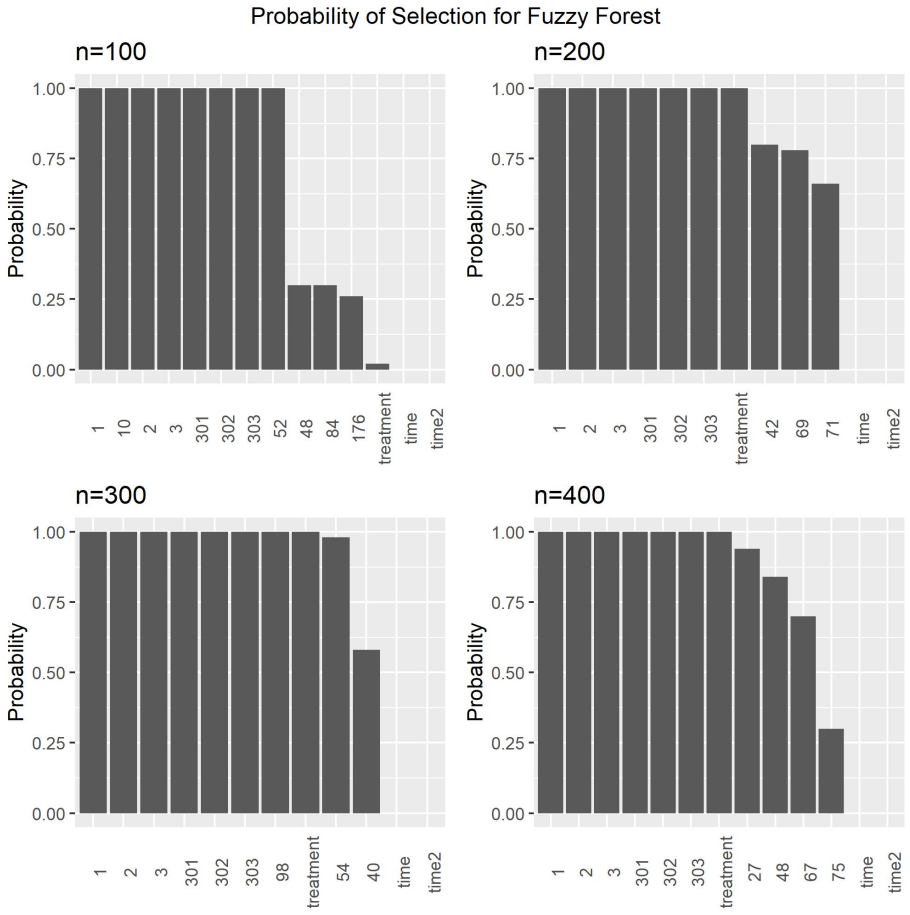

Fig. 4: Feature selection performance of Fuzzy Forests on the dataset with a time-treatment interaction

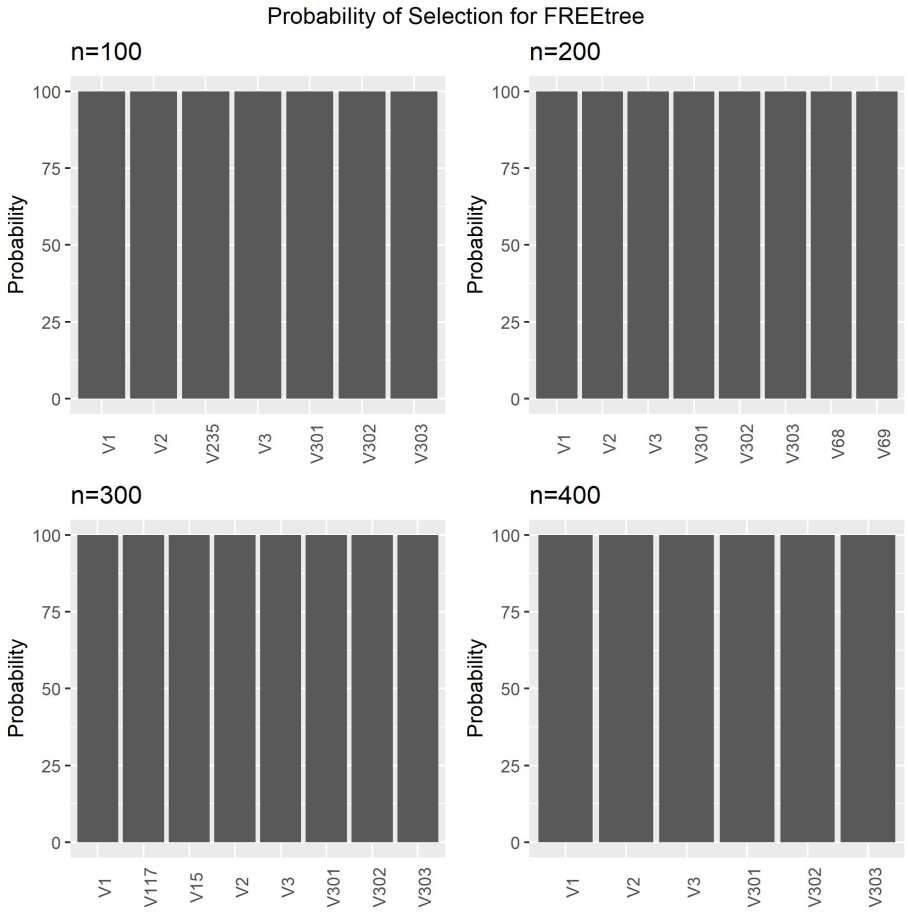

Fig. 5: The selected feature of FREEtree with different sample size n on the dataset with only random intercepts. The first column is the number of patients or sample size n.

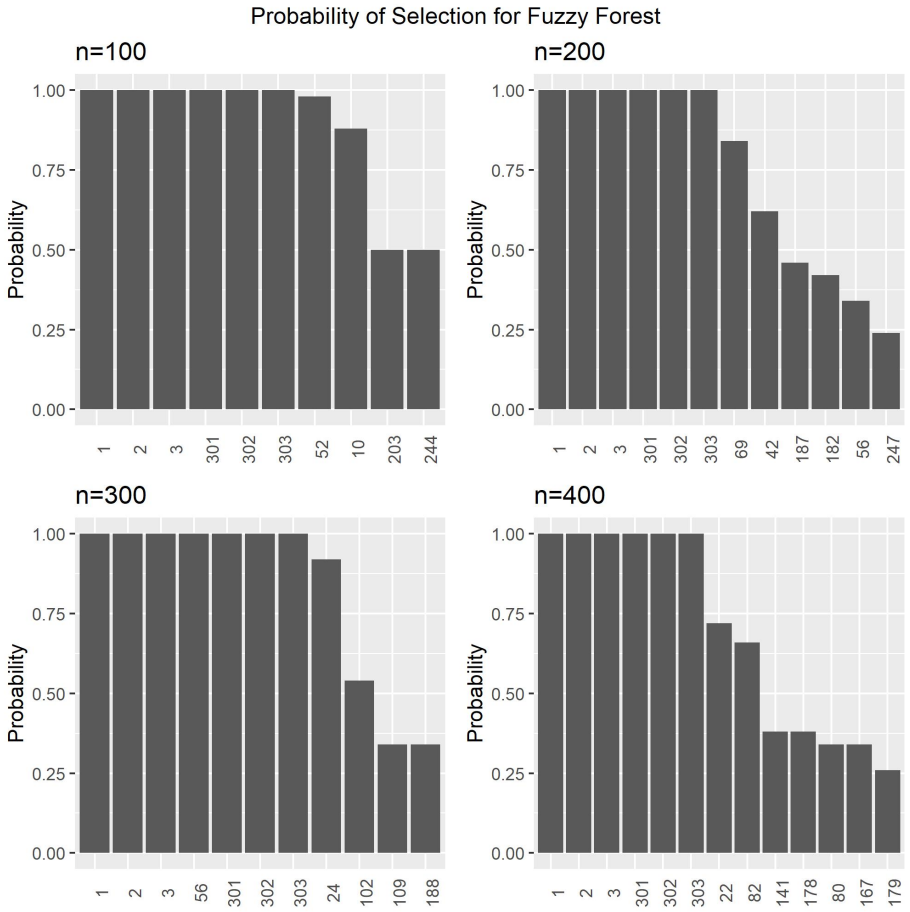

Fig. 6: Feature selection performance of Fuzzy Forests on the dataset with only random intercepts

4.4 Estimation of the underlying pattern

The advantage of FREEtree not only lies in higher prediction accuracy, but also in fitting the underlying structure due to the models at its leaves. Recall that in the first simulation, dataset has a time-treatment interaction. That is, the treatment-time components will first drop then increase for treatment1 and will first increase and then drop for treatment2. In this section we will examine whether FREEtree can recover the true time pattern for different treatments. The underlying true pattern should have the following form:

$$\begin{cases} (t-3)^2 & \text{treatment} = 1 \\ -(t-3)^2 & \text{treatment} = 2 \end{cases}$$

FREEtree was able to successfully detect the time-treatment interaction in this simulation. Table 1 shows that FREEtree gives a reasonable estimation of the time pattern function. However, note that patterns like this cannot be directly observed using tree-based methods such as RE-EM tree because the leaves in RE-EM tree correspond to an averaged value instead of a model.

Sample Size	treatment1		treatment2	
	time	time2	time	time2
100	−8.88	1.36	5.23	−0.89
200	−5.60	0.88	5.46	−0.91
300	−6.06	0.99	5.43	−0.91
400	−6.40	1.07	6.16	−1.01

Tab. 1: The mean of coefficients of linear models at leaves for each treatment. The coefficients of time and time2 should be 6 and 1 for treatment1 and -6 and -1 for treatment2.

5 Application

We illustrate a real data application of FREEtree in a wide longitudinal dataset of World Bank, IMF and Penn World Table country level economic and developmental indicators. Using the adoption of inflation targeting by a nation's central bank as a treatment variable, we wish to predict the percentage change in a country's consumer price index (CPI) as a measure of the inflation rate. Merging together 15 different data sources[1], we obtain a final data set of 120 countries with 393 features observed for a 12 year period between 2005 and 2016 inclusively. The data series mostly comprise of population ratios, per capita metrics, year-over-year rates of change, proportions of national accounts, and scaled indicators, before being normalized to have mean zero and unit standard

[1] IMF World Economic Outlook (October 2019), IMF Financial Development Index Database, Penn World Table version 9.1, and the following World Bank databases: World Development Indicators, Education Statistics, Doing Business, Health Nutrition and Population Statistics, Gender Statistics, Global Financial Development, Health Equity and Financial Protection Indicators, Worldwide Governance Indicators, Worldwide Bureaucracy Indicators, Statistical Capacity Indicators, Global Jobs Indicators and Environment, Social and Governance Data.

deviations.

Country level indicators are often highly correlated across time, with many series being very related to or subsets of others. Although tree-based techniques like Random Forests and Fuzzy Forests can process large numbers of series through feature selection, they do not have the capabilities to model mixed effects or give a single interpretable tree. **glmertree** can manage such effects while directly incorporating treatment variables into the analysis using GLM at each final node, it cannot handle the number of features in the dataset given the dimensionality problems inherit in linear regression. We compare results obtained by FREEtree to Random Forests and Fuzzy Forests in an example that takes advantage of individual country-level effects, as well as the central bank price targeting policy *country_id* was declared to be the subgroup **cluster**, while **fixed regress** included a linear and quadratic temporal term. Inflation targeting adoption, a binary variable, was declared for **consider split**, the rest of the features were included for the screening and selecting process. The formula takes the form:

$$CPI_{i,t} = year_t + year_t^2 + treatment_{i,t} + X_{i,t} \mid country\_id_i \mid X_{i,t}$$

where $X_{i,t}$ include all the other features to be screened and selected by FREEtree's algorithm.

The resulting tree has three nodes from two split variables (investment price index and GDP volatility) and 9 explanatory variables, including GNP per capital, fuel and GDP volatility (Figure 7). The mixed effect paint a picture of volatile frontier economies in various states of high inflation or deflation, while industrialized nations tend to be closer to the mean (Figure 8).

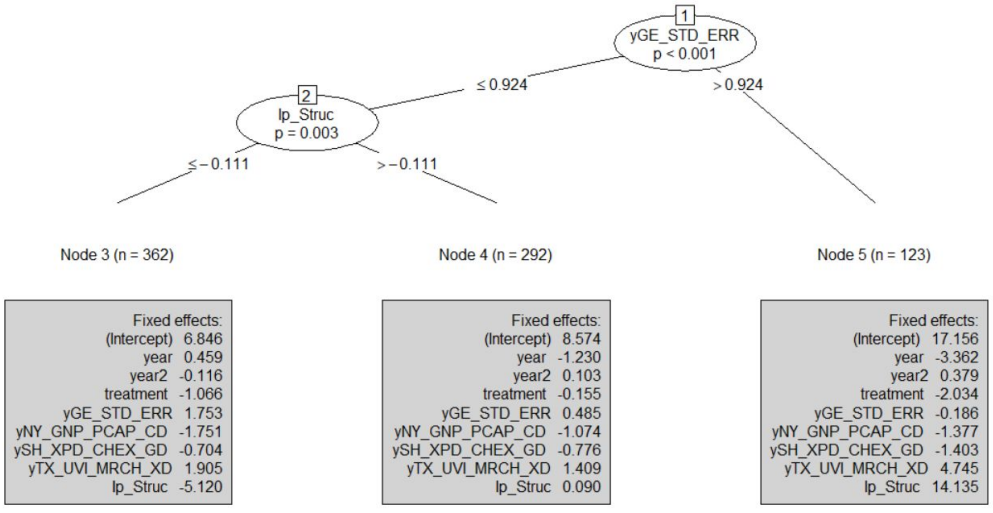

Fig. 7: FREEtree model tree applied to real sample data

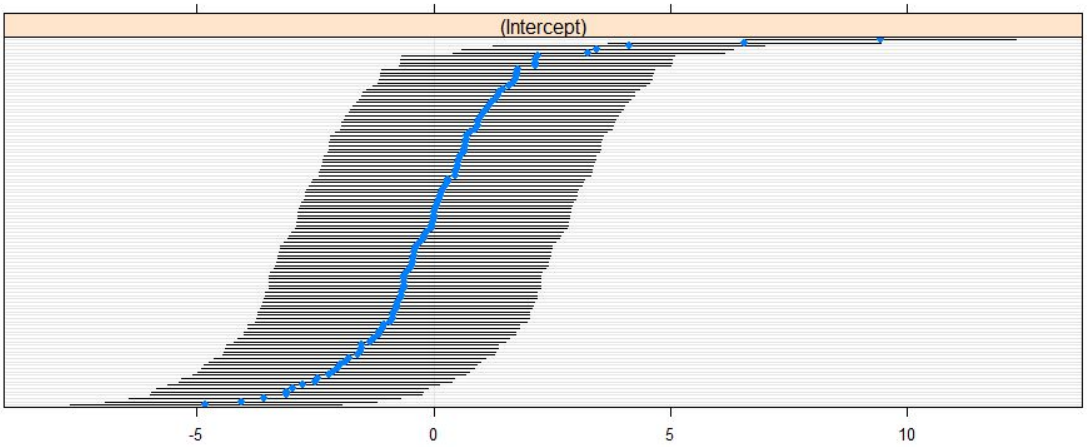

Fig. 8: FREEtree model tree applied to real sample data: individual country effects

Using mainly default parameter values, WGCNA yielded four modules with 150, 125, 80 and 38 features, and the grey module being the 3rd largest in size. We can see that FREEtree performs notably better in larger samples (Figure 9) and further out of sample temporally (Figure 10).

6 Interpretability

FREEtree differentiates itself from other model trees in its ability to accept a very large number of features, addressing dimensionality issues when $p >> n$. Although this is a feature it shares in common with Random Forest and Fuzzy Forest, it distinguishes itself from these ensemble methods by being able to produce a single tree the user can readily interpret and understand while also providing superior predictions.

The functionality in a single decision tree over popular forest ensemble methods without sacrificing performance also includes the ability for the user to specify persistent features that will make it into the regression nodes, inherited from LMM tree. In addition, the user can specify subgroup cluster indicators and which features are guaranteed to make it passed the screening process as regressors in the linear model. This flexibility caters well to researchers seeking to understand the impact of their variables of interest among a high amount of other features, allowing them to effectively customize the output tree while taking advantage of WGCNA-based feature selection.

Random Forest has to run multi-node decision trees many levels deep (Figure 11) recursively to achieve comparable performance to FREEtree's 5 node fixed-effect regression tree with 9 explanatory variables (Figure 7).

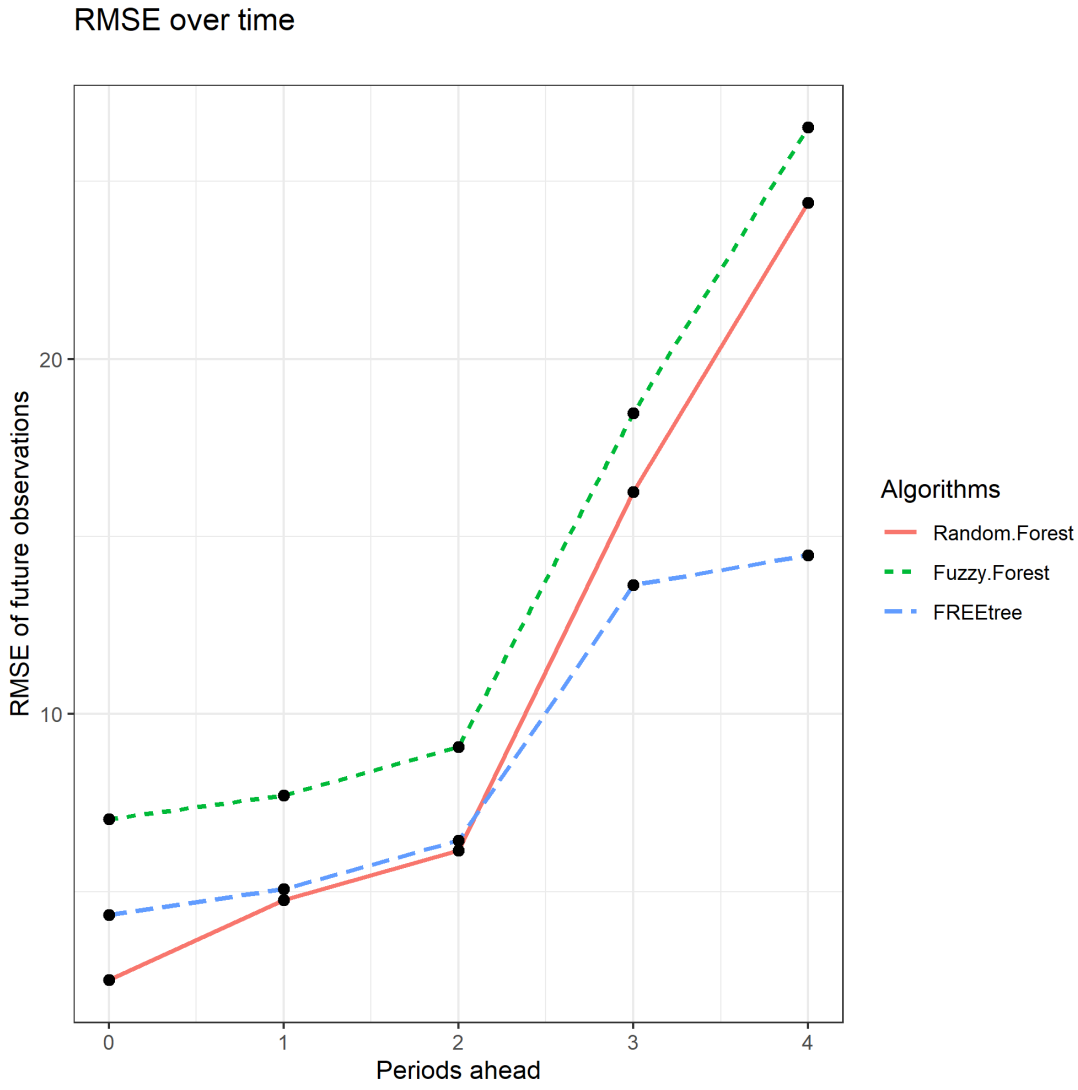

Fig. 9: FREEtree model tree applied to real sample data: cross sectional performance (20 test countries)

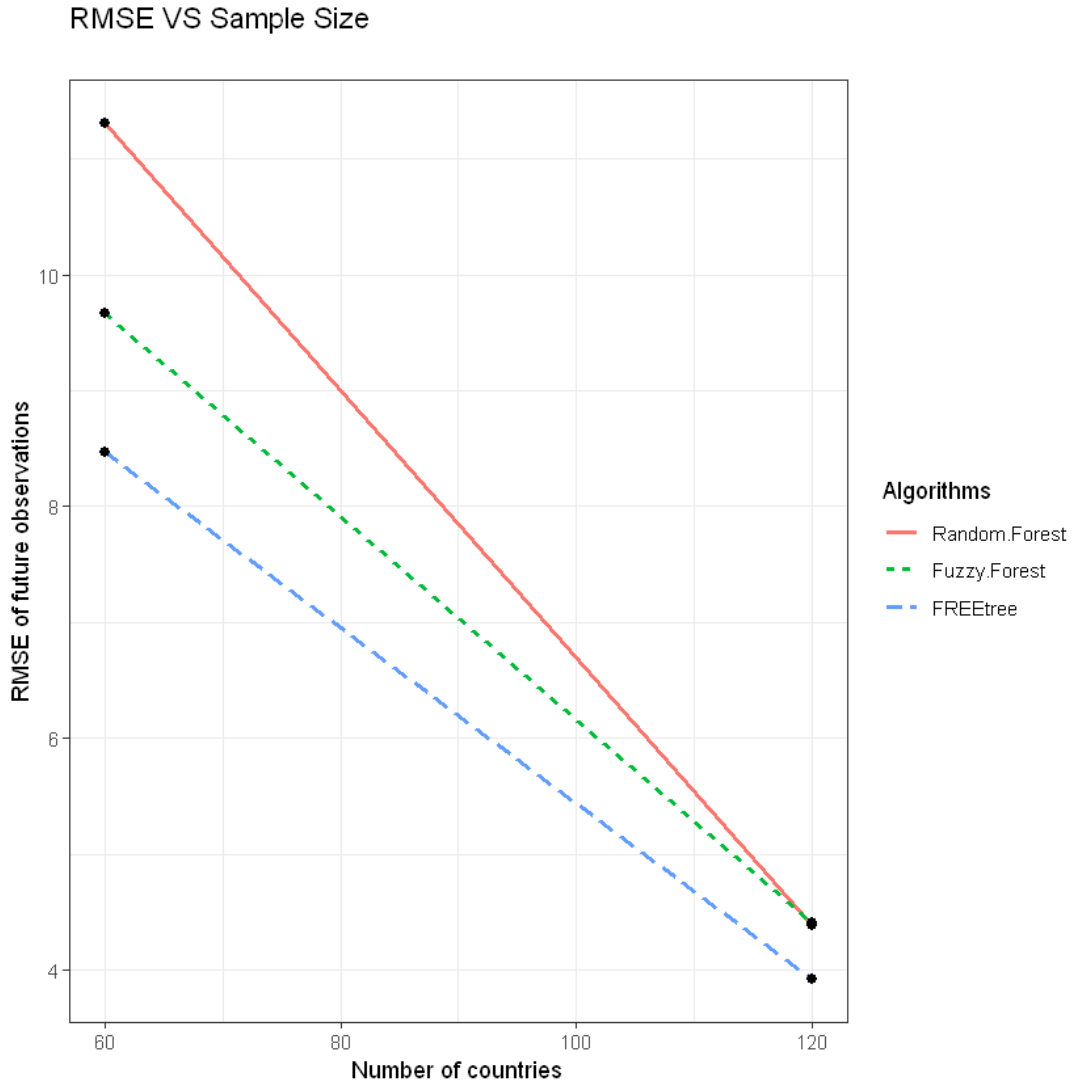

Fig. 10: FREEtree model tree applied to real sample data: forward performance by horizon

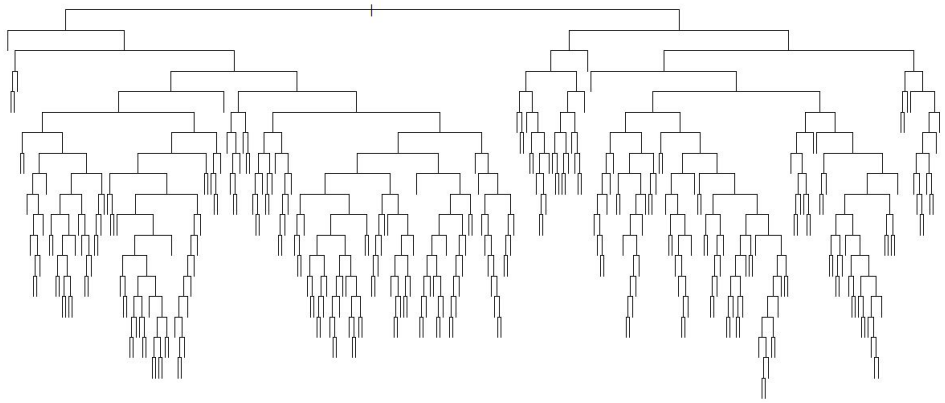

Fig. 11: Representative tree of Random Forest applied to real sample data

7 Discussion and Future work

At the feature clustering step, FREEtree uses Pearson correlation as the similarity function, which may not be optimal when the measurements of each feature of any patients are time series. That is, for patient i and feature v, $X_{i1}^{(v)}, X_{i2}^{(v)}, .., X_{iT}^{(v)}$ is a time series. In order to cluster features in this case, we have to cluster time series. According to our simulations, where Auto-Regressive and Compound-Symmetric structure were imposed on each feature $X^{(v)}$, WGCNA still works when the correlation between features are relatively large. However, when correlation is relatively low, WGCNA may not find strong association and assign all the features to grey group. One way to get around this is that when doing WGCNA analysis, instead of using correlation of features when building similarity matrix, we use time series distance measure such as Dynamic time warping (DTW) and average them with respect to each patient and finally transform it into a similarity measure. In this case, the adapted WGCNA can detect module distinctions even if the correlation between features is relatively low. However, it is most be pointed out that computing time series distance measure such as DTW requires a lot of computational resources and in applications where p is really large, replacing correlation with a time series distance measure may not be practical computationally.

8 Conclusions

In this paper we have presented Fuzzy Random Effect Estimation tree (FREEtree) algorithm that can provide a relatively unbiased way to do feature selection in the presence of correlation between features. Also, it deals with longitudinal data by using random effect model tree, where the fixed effect is modelled as

a piecewise linear model, which has greater fitting and predicting power than RE-EM tree. It is expected that FREEtree can be widely used in application where the data has longitudinal structure as well as many correlated features.

References

[1] Leo Breiman, Jerome H Friedman, Richard A Olshen, and Charles J Stone. Classification and regression trees. belmont, ca: Wadsworth. *International Group*, 432:151–166, 1984. 1

[2] Daniel Conn, Tuck Ngun, Gang Li, and Christina Ramirez. Fuzzy forests: extending random forests for correlated, high-dimensional data. 2015. 1

[3] Ramón Díaz-Uriarte and Sara Alvarez De Andres. Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7(1):3, 2006. 1

[4] Soo-Heang Eo and HyungJun Cho. Tree-structured mixed-effects regression modeling for longitudinal data. *Journal of Computational and Graphical Statistics*, 23(3):740–760, 2014. 1

[5] Marjolein Fokkema, Niels Smits, Achim Zeileis, Torsten Hothorn, and Henk Kelderman. Detecting treatment-subgroup interactions in clustered data with generalized linear mixed-effects model trees. *Behavior research methods*, 50(5):2016–2034, 2018. 1, 2.2, 2.2

[6] Wei Fu and Jeffrey S Simonoff. Unbiased regression trees for longitudinal and clustered data. *Computational Statistics & Data Analysis*, 88:53–74, 2015. 1

[7] BL Gudenas and W Liangjiang. Gene coexpression networks in human brain developmental transcriptomes implicate the association of long non-coding rnas with intellectual disability. *Bioinform Biol Insights*, 9(Suppl 1):21–27, 2015. 2.1

[8] Torsten Hothorn, Kurt Hornik, and Achim Zeileis. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical statistics*, 15(3):651–674, 2006. 1

[9] S.S. Kim, R.M. Alvarez, and C.M. Ramirez. Who voted in 2016? using fuzzy forests to understand voter turnout. *Social Science Quarterly*, 2020. 1

[10] D Langfelder and Horvath S. Wgcna: An r package for weighted correlation network analysis. *BMC bioinformatics*, 9(1):559, 2008. 2.1

[11] G Pei, L Chen, and W Zhang. Wgcna application to proteomic and metabolomic data analysis. *Methods Enzymol*, 585:135–158, 2017. 2.1

[12] C.M. Ramirez, M.A. Abrajano, and R.M. Alvarez. Using machine learning to uncover hidden heterogeneities in survey data. *Scientific Reports*, 9, 2019. 1

[13] Mark Robert Segal. Tree-structured methods for longitudinal data. *Journal of the American Statistical Association*, 87(418):407–418, 1992. 1

[14] Rebecca J Sela and Jeffrey S Simonoff. Re-em trees: a data mining approach for longitudinal and clustered data. *Machine learning*, 86(2):169–207, 2012. 1, 3.4

[15] Achim Zeileis, Torsten Hothorn, and Kurt Hornik. Model-based recursive partitioning. *Journal of Computational and Graphical Statistics*, 17(2):492–514, 2008. 2.2

[16] Bin Zhang and Steve Horvath. A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology*, 4(1), 2005. 1, 2.1, 2.1