Humboldt Universitat zu Berlin

Generelized Random Forest

Malgorzata Paulina Olesiewicz

598939

July 10, 2019

Contents

List of Figures

List of Tables

1 Motivation

Randome forest has been first introduced by Breiman in 2001 and became a very popular Machine Learning prediction technique. The foundations of the method have been already discussed a decade earlier with Ho introducing the term "random forest" in 1995.The basic principle of the technique is the minimization of the variance of the prediction though taking an average of many "noisy" but unbiased models (Hastie et al. 2001).
Since its creation random forest has been developed for number of different applications, focused on the type of data and the outcomes needed to be evaluated. This paper will focus on the Generalized Random Forest (GRF) developed by Susan Athey, Julie Tibshirani and Stefan Wager over the last 3 years. The method applies the concept of Generalized Method of Moments for the purpose of better prediction of herteroskedastic coefficients, particularly focused on non-parametric data, partial causal treatment effect and instruments.
In order to present the contribution of the paper to the field, first the basic methods used in the random forest will be discussed. The "classical" approach will be then compared with the method developed by Athey and al. For this purpose motivation of both methods, their assumptions and steps of the algorithms will be considered. Finally, the example of the quantile applying the method will be discuss and conclusion regarding the approach will be given.

2 Preliminaries

2.1 M- estimator

The bases of the coefficient estimation in the Generalized Random Forest is the M-estimator, which has been proposed by Peter J. in 1964 as generalized approach to maximum likelihood estimation of parameters of non-linear models. In many applications, such M-estimators can be thought of as estimating characteristics of the population. M-estimation includes among others maximum likelihood, nonlinear least squares, least absolute deviations, quasi-maximum likelihood (Wooldridge 2001).Therefore, the estimator allows for the flexibility required for application of the Generelized Random Forest to vast range of models. Since we are assuming non-linearity of the parameters, certain assumptions regarding their space needs to satisfied:

- Θ is a compact subset of R^P
- $q(w, \theta)$ is continuous in θ for all w
- function q is Borel measurable

We will come back to those assumptions while discussing the set up of the Generelized Random Forest. The objective of the M-estimator θ_{o} is to minimize equation:

$$\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} N^{-1} \sum_{i=1}^{N} q\left(\mathbf{w}_i, \boldsymbol{\theta}\right) \tag{1}$$

where q is a a squared difference between data y and the model prediction: $q(w, \theta) = \left[y - m\left(\mathbf{x}, \boldsymbol{\theta}\right)\right]^2$. Therefore , M-estimator is often through of as a non-linear least squares estimator.The parameter of interest θ_0 is the value of θ that uniquely solves the population problem given by:

$$\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathrm{E}[q(\mathbf{w}, \boldsymbol{\theta})] \tag{2}$$

To solve the minimisation, the moment condition needs to be satisfied:

$$\mathrm{E}\left[\mathbf{s}\left(\mathbf{w}, \boldsymbol{\theta}_{\mathrm{o}}\right)|\mathbf{x}\right] = -\nabla_{\theta} m(\mathbf{x}, \boldsymbol{\theta})' \left[\mathbf{E}(y|\mathbf{x}) - m\left(\mathbf{x}, \boldsymbol{\theta}_{\mathrm{o}}\right)\right] = \mathbf{0} \tag{3}$$

Since the M-estimator solves the population problem, is can be solved for by using the analog principle and we can replace its estimation with the sample average. Moreover, we can assume its asymptotic consistency:

$$N^{-1} \sum_{i=1}^{N} q\left(\mathbf{w}_i, \boldsymbol{\theta}\right) \xrightarrow{p} \mathrm{E}[q(\mathbf{w}, \boldsymbol{\theta})] \tag{4}$$

The sample estimator $\hat{\theta}$ minimizes the left side of equation 4 whereas θ_{o} minimizes the righ-hand side. Therefore, it may be assumed. $\hat{\boldsymbol{\theta}} \xrightarrow{p} \boldsymbol{\theta}_{\mathrm{o}}$.

Through Tailor approximation it is possible to derive the normality and consistency of the M-estimator as well as its asymptotic variance allowing us for construction of confidence intervals and inference. This will be the crucial point while discussing the statistical validity of the Generalized Random Forest. The last point, which needs to be highlighted here is the fact that method of moments estimators tend to be bias.

2.2 Decision Tree

The method of random forest is heavily based on the concept of decision tree which is often refereed to as Classification and Regression Tree (CART). Decision tree was also Athey's starting point in the efforts to combine machine learning and econometric techniques. This section will discuss the concept of decision tree and Athey's contributions to this field, which will be relevant for the further discussion on her version of the random forest. Decision tree classifies data according to the observations' characteristics. The underling idea here is that observations with similar characteristics will most likely have similar outcomes. The tree is developed though recursive partitioning where observations are classified into subsamples, each of them with different outcome which can be a category (in case of categorical data) or constant (in case of regression). For latter, each outcome can be expressed as:

$$\hat{c}_m = \mathrm{ave}\left(y_i | x_i \in R_m\right) \tag{5}$$

Where $\hat{c}_m$ is just an average of y_i in the region (node) R_m. The covariate with the biggest predication leverage from all available covariates is chosen as the first node (root node) of the tree. The data from the parent node are split into two resulting regions in such way to minimize the prediction error of the outcome. The ideal split s for variable j will satisfy classical sum of squares equation:

$$\min_{j,s} \left[\min_{c_1} \sum_{x_i \in R_1(j,s)} \left(y_i - c_1\right)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} \left(y_i - c_2\right)^2 \right] \tag{6}$$

The computation is repeated with the greedy algorithm for every split combination and the best pair (j, s) is chosen. For categorical data the split is determined through comparison of Gini impurity, where again the best split will offer the smallest error of prediction. As it will be shown, in GRF the split rule will be defined quite differently.

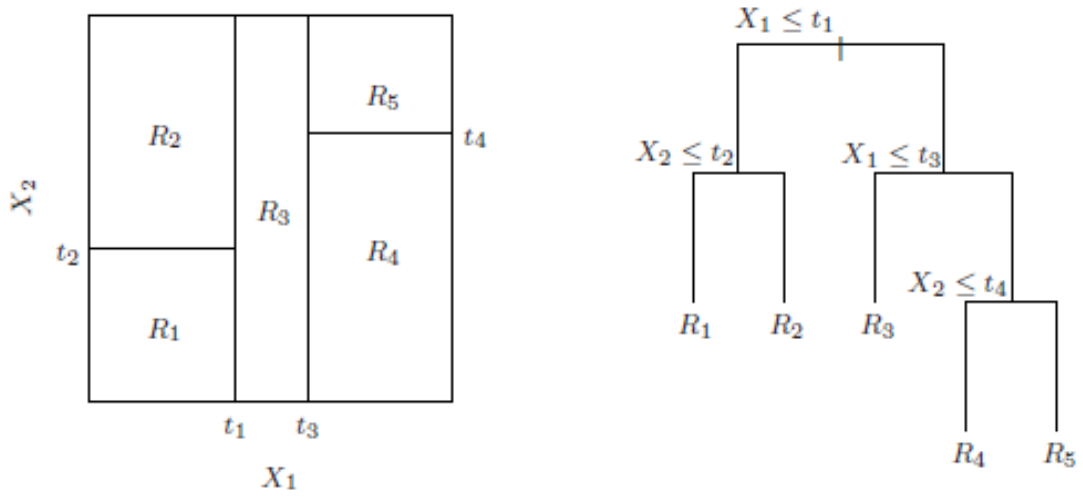

Figure 1: *Decision Tree.*
Left: partition of a two dimensional feature space Right: corresponding decision tree. (Hastie et al. 2001)

Once the data has been split into two groups, the resulting nodes (children in Athey's notation) become parents and another variable with the second biggest prediction leverage is chosen to form the rule for the split. The objective is to find splits which allow to immediately improve the quality of the prediction as much as possible (Athey et al. 2019).Figure 1 is an example of decision tree. The final outcomes (nodes without children) are called the leaves nodes and in the literature it has been agreed that they should have no less than 5 observations (Hastie et al. 2001), otherwise we are risking overfitting. Decision trees are excellent tool for building the models explaining the outcomes in the observed data, however due to their high variance, they are known not to perform well for prediction.

2.3 Boosting and Influence Function

The boostrap method was developed to asses the uncertainty of estimation though sub-sampling of the data. The final estimator of the coefficient is derived by taking an average of a number of estimators obtain from different sub-samples of the training data. One can obtain an estimator from the sample by fitting a cubicspline to the data. Example of this operation can be seen in figure 2 where we split our training data into seven sub-samples. The space of function can be represented by the B-spline basic functions:

$$\mu(x) = \sum_{j=1}^{7} \beta_j h_j(x) \tag{7}$$

where our $\mu(x)$ represents the conditional mean of Y on X. The estimation of the $\hat{\beta}$ coefficient can be executed by minimizing the square error, known to us from the OLS estimation. The $\hat{\mu}(x) = \sum_{j=1}^{t} \hat{\beta}_j h_j(x)$ will represent then the fitted estimation of the conditional mean and is shown on the third diagram of Figure 2.

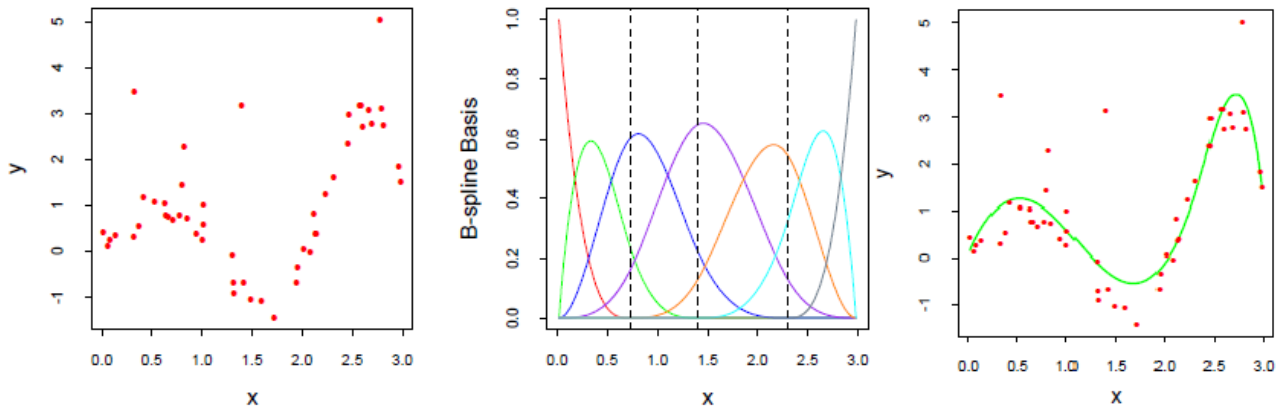

Figure 2: *Boostraping*

Left: distribution of the data Middle : fitting 7 cubic splines to the data Right: boostrap estimate for the data based on the 7 seven sub-samples. (Hastie et al. 2001)

The boostrap method gives the same result as the maximum likelihood estimation and is valuable alternative in non-parametric setting. Often, the disadvantage of the maximum likelihood is that it requires the knowledge of number of assumptions and availability of formulas. Therefore, the boostrap maximum like-lihood estimation is preferred Hastie et al. (2001).The boostrap method allows us to estimate a stable estimator based on the data sample. However, another aspect we may be interested in is how different observations impact the estimation of the coefficient. For this purpose we can use influence function.

The empirical influence function is a measure of the dependence of the estimator on the value of one of the points in the sample. It is a model-free measure in the sense that it simply relies on calculating the estimator again with a different sample. One of the ways of estimating the influence of the observation is so-called cook distance. Here we look at the coefficient estimated without the observation i (β_i)and the estimate with the observation β. The calculation is conducted as follow:

$$CD_i := \left(\hat{\boldsymbol{\beta}}_{(i)} - \hat{\boldsymbol{\beta}}\right)' \widehat{F(\hat{\boldsymbol{\beta}})} \left(\hat{\boldsymbol{\beta}}_{(i)} - \hat{\boldsymbol{\beta}}\right) \tag{8}$$

The cook distance is very useful tool to identify the observation which are the most influential for the estimation of the parameter. As it will be shown later, it will also help to determine the heterogeneity of the parameter based on observation's influence in estimating the parameter of the parent nod.

2.4 k-Nearest-Neighbor Classifier

The k-Nearest-Neighbor classifiers are memory-based and do not require to fit specific model, which makes it perfect for non-parametric data. Given a query point x_0, k training points which are the closest in distance are found, and then classify using majority vote among the k neighbors (Hastie et al. 2001). For simplicity it is often assumed that the features are real-values and we are operating in Euclidean space:

$$d_{(i)} = \left\|x_{(i)} - x_0\right\| \tag{9}$$

Consequently, the kernal weighting function puts more weight on the observations closer to the original data point, however if we increase the dimensions of our model we may run into the issue often discuss as "curse of dimensionality". In Athey's random forest k-Nearest-Neighbor Classifier will group the observations based on whether they up in the same end node or not. Therefore, trees and forests can be taught of as nearest neighbor method with adaptive neighborhood metric, where the closeness is

defined with respect to decision tree. The adaptive refers to the change of distance in accordance to which tree we are referring to in the forest. For details see section 4.2

3 Random Forest

Random forest is an adaptation of the decision tree methodology for the purpose of prediction. Through taking an average of a high number of bias but highly variable estimates, the variance is significantly reduced and consequently a stable prediction is achieved. In this section the basic concepts used in the random forest will be explained and the description of random forest algorithm will be given.

3.1 Bagging and the basic Algorithm

Bagging (bootstrap aggregation) focuses on the redaction of variance of the prediction through randomization and averaging the predictions of many unbiased but noisy models. The first step of the process is to create a subsample Z of the same size (N) as the original data set by random drawing observation with replacement. The subsample of data, which has not been selected is called "out-of-bag" and is used for testing of the tree. Once the subsample is created, the covariate for the root note is chosen randomly. The subsample of covariates available for building the tree vary every time. Then the boostrap tree is build as before through recursive partitioning, however after every split the next covariate is chosen randomly from the remaining in the model. Figure 3 presents example of original decision tree and five corresponding boostrap trees. Each boostrap tree has a different root note, different splits are used and different number of leaves are created.

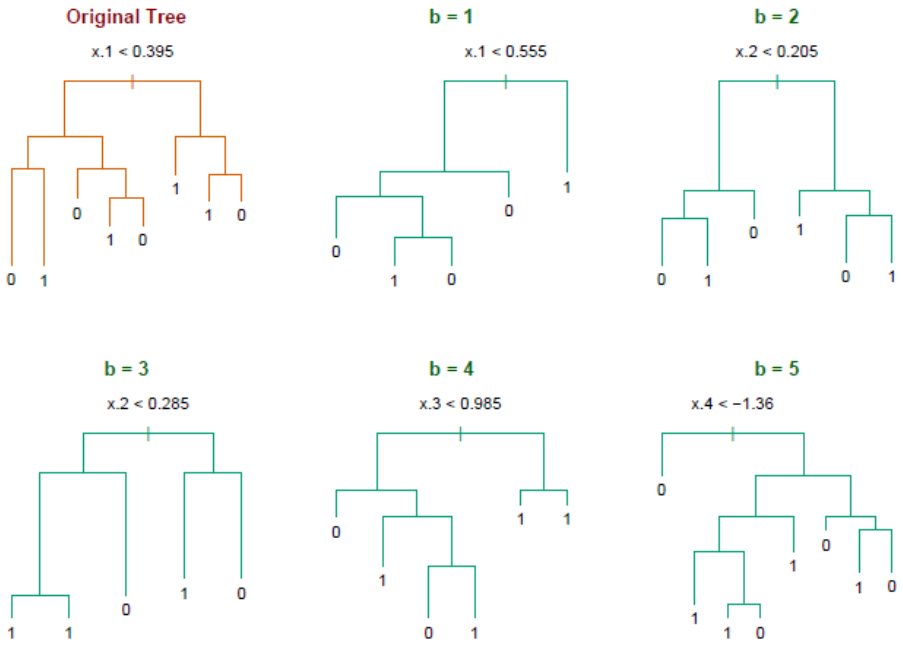

Figure 3: *Bagging*
Different trees are build from different subsamples. For every root node is different. (Hastie et al. 2001)

Based on all the estimators of boostrap trees B (usually somewhere between 200-300) the bagging estimate can be derive as a Monte Carlo estimate of the true bagging estimate, approaching its true

value as $B \to \infty$. Consequently we obtain the average of all the estimates:

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(x) \tag{10}$$

Conventional CART algorithm composes of two parts: a) building a tree which by construction will be over-fitted and b) cross-validation to select a penalty of the tree depth. Each part of the algorithm relies on a criterion function based on mean-squared error. In the second stage the cost of each leaf is evaluated to "prune" the tree. Since the goodness of fit criterion for the Mean Square Error can be expressed as:

$$-\text{MSE}\left(\mathcal{S}^{\text{tr,cv}}, \mathcal{S}^{\text{tr,tr}}, \Pi\right) \tag{11}$$

from which it can be seen immediately that since we are using two different samples the too-extreme estimates leaves will be penalized. The smaller leaves are directly indicating deeper tree with noisier estimates and higher average MSE for the cross-validation samples (Athey & Imbens 2016).
In terms of algorithm the steps of the Random Forest Algorithm can be represented as:

Random Forest Algorithm
1. For b = 1 to B:
(a) Draw a bootstrap sample Z of size N from the training data
(b) Grow a random-forest tree Tb to the bootstrapped data, by re- cursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
i. Select m variables at random from the p variables.
ii. Pick the best variable/split-point among the m.
iii. Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_b\}_1^B$
To make a prediction at a new point x:
Regression: $\hat{f}_{\text{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$ Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $C_{\text{rf}}^B(x) =$ majority vote $\{C_b(x)\}_1^B$

Table 1: *Random Forest Algorithm*

Sources: Hastie et al. (2001) page 588.

3.2 U-Statistics

U-statistics is a commonly accepted notions of unbiased estimation such as the sample mean and the unbiased sample variance (the "U" stands for "unbiased") in a large sample models. Formally the U-statistics can be written as:

$$U_n = \frac{1}{\begin{pmatrix} n \\ a \end{pmatrix}} \sum_{1 \le i_1 < \cdots < i_a \le n} \phi\left(X_{i_1}, \ldots, X_{i_a}\right) \tag{12}$$

where $\phi(x_1, \ldots, x_a)$ stands for kernel weighting function associated with the expectation functional $\text{T(F)} = \text{E}_F \phi(X_1, \ldots, X_a)$ (expectation of the density of the kernel function) and where n is the sample size. The unbiassnes of U_n is associated with averaging $\begin{pmatrix} n \\ a \end{pmatrix}$ where each term is associated with expectation functional T(F). Obviously, we assume our random variable to be i.i.d distributed Yuan et al. (2017).
Rather than weighting all the trees equally one can use kernel weighting function to account for the

diversity of the predictions for given observation. This allows the analysis of the final estimation using the U-statistics, which has been defined in the literature and we know how to test statistical significance of the results.

3.3 Prediction vs Casual Inference

A model estimated using the traditional least square approach is often overfit and leads to very poor prediction in the out-of-sample setting. If, for instance, model has exactly the same number of coefficient as data observations, the least squares perfectly fits the model and returns the R^2 of one, which will obviously lead to very poor prediction. This is mainly due to the fact that the least squares captures not only the information regarding the variation of the outcomes based on the variables but also the noise within the sample used for the estimation (Leeb & Pötscher 2008).

The suggested in the literature solution to this issue is related to regularization of the estimation through reduction of the number of the coefficients, which need to estimated.This "approximately spare" regression model Often uses LASSO or other regularization methods to identify coefficients essential for the prediction. Therefore, the assumption of "sparsity" often must be imposed in order to guarantee consistency and asymptotic normality of the estimator. The difficulty in drawing inferences after regularization or model selection is that these procedures are designed for forecasting, not for inference about model parameters. One solution would be to focus on the model selection rather than predictive parts.

Random forest as well as other machine learning techniques are very good at the prediction of models' outcomes. The approach uses separate subsamples to build the model and check for its accuracy,resulting in highly accurate prediction. However, if we use random forest for causal inference we will run into the issue of building the model (growing the tree) and obtaining the parameters from the same data. This is not an issue in prediction, since leaves simply report the average of the observations, whereas in casual inference the derivation of the estimators is done with the same observations which was used for partitioning. Consequently, this may lead to incorrect conclusion and invalid inference of the model.

The other issue with casual inference is that we do not really observe the "ground truth" of our causal effect, which is often call the "fundamental problem of causal inference" (Athey & Imbens 2016). In other word, it is not possible to conduct cross-validation in the traditional way. Therefore, often minimizing the Mean Square Error does not make sense as a criterion for model goodness of fit. For further discussion on this topic please see Leeb & Pötscher (2008).

3.4 Honest Forest

The traditional CART approach does not allow for constructing the confidence intervals, necessary for casual inference. Moreover, the model uses training data for model selection and estimation of the coefficients; therefore the spurious correlation (not causal) between covariates and outcomes leads to a bias estimation (Athey & Imbens 2016). As mentioned before, one solution to this issue would be to restrict the model. Another way could be application of the method for recursive partitioning for heterogeneous casual effects developed in 2015 by Athey and Imbens, which strongly advocates usage of two different subsamples for partitioning and estimation step. The method also does not impose any restrictions on the number of coefficients contributing to prediction of the outcome. The so-called

"honest" CART redefines the Mean Square Error criterion for the partitioning Π as:

$$\text{MSE}\left(\mathcal{S}^{\text{ te}}, \mathcal{S}^{\text{ est}}, \Pi\right) \equiv \frac{1}{\#\left(\mathcal{S}^{\text{ te}}\right)} \sum_{i \in \mathcal{S}^{te}} \left\{ \left(Y_i - \hat{\mu}\left(X_i; \mathcal{S}^{\text{ est}}, \Pi\right)\right)^2 - Y_i^2 \right\} \tag{13}$$

Where S^{te} stand for the test sample and S^{est} for the estimation sample. The estimated coefficients in each leaf is unbiased as the sample used for the partitioning is different than for its estimation. This way, the honest tree is eliminating one aspect of over-fitting. The ultimate goal of the algoritms $\pi(.)$ used on the training sample $S^{te}$is to maximize the honest criterion:

$$Q^H(\pi) \equiv -E_{\mathcal{S}^{\text{ est}}, \mathcal{S}^{\text{ est}}, \mathcal{S}^{\text{ est}}} \left[\text{MSE}\left(\mathcal{S}^{\text{ te}}, \mathcal{S}^{\text{ est}}, \pi\left(\mathcal{S}^{\text{tr}}\right)\right)\right] \tag{14}$$

However, the issue arises since we do not observe the "true" parameter of interest or treatment effect and therefore the MSE needs to be estimated. In other words, it is not possible to conduct normal cross-validation as the testing data is not able on its own to examine the accuracy of the model Athey & Imbens (2016). The estimated MSE for the split can be expressed by:

$$\widehat{\text{EMSE}}\left(\mathcal{S}^{\text{tr}}, \Pi\right) \equiv \frac{1}{N^{\text{tr}}} \sum_{i \in \mathcal{S}^{\text{tr}}} \hat{\mu}^2\left(X_i; \mathcal{S}^{\text{tr}}, \Pi\right) - \frac{2}{N^{\text{tr}}} \cdot \sum_{\ell \in \Pi} S^2_{\mathcal{S}^{\text{tr}}}(\ell) \tag{15}$$

where both sample size of estimating and testing sample are this same size and therefore we obtain term $\frac{2}{N^{\text{tr}}}$ and $S^2_{\mathcal{S}^u}(\ell)$ denotes within-leaf variance. Despite the fact that the EMSE is approximately unbiased for specific partitioning Π, it is biased when used repeatedly to evaluate splits using recursive partitioning on the training data S^{tr}. The variance of training data in the leaf tends to be smaller than the sample variance would be in a new, independent sample since the initial split groups the observations with similar extreme outcomes Athey & Imbens (2016). Therefore, cross validation still plays important role in the honest decision tree. Consequently, the honest criterion, which needs to be satisfied in equation 15, can be evaluated using cross-validation sample $-\widehat{\text{EMSE}}\left(\mathcal{S}^{\text{tr,cv}}, \Pi\right)$. The estimator will be unbiased but may have high variance due to a small amount of observation for cross-validation.

The approach of honest tree has also been applied to the random forest. The honest tree derived by Athey & Imbens (2016) focued on developing single well-tuned tree with fairly large leaves, which would ensure the accuracy of the estimation. In random forest, the leaves can be much smaller as the final estimation is based on averaging of many unbiased but highly variant models. Originally, the honest random forest was developed for prediction of casual treatment effect rather than unit's outcome. With the honest approach the random forest is proven to achieve both consistency and centered asymptotic normality, which is necessary for the statistical inference and was not possible with standard bagging approach.

We discussed all the concepts, which are the foundations of the random forest algorithm. The steps of the standard random forest algorithm and potential issue with its adaptation for the purpose of prediction have been highlighted. Now we are ready to discuss the Generalized Random Forest approach introduced in the paper.

4 Generelized Random Forests

Susan Athey published her first paper on random forest in 2015, where she used the leanings from her research on decision tree and applied it for the purpose of forest based prediction of partial treatment effects. The *Generalised Random Forest* paper from 2019 co-written with Tibshirani and Wager describes the latest developments of the method which now can be adopted to wider range of applications. The paper was accompanied with the release of beta version of the R package grf. In this section, the key elements of GRF algorithm will be discussed, followed by summary of the method's asymptotic properties and comparison with the classical approach.

4.1 Overview and Assumptions

The main objective of the method is to estimate parameters of interest via local moment conditions. Given specific data $(X_i, O_i) \in \mathcal{X} \times \mathcal{O}$ the method seeks to estimate highly heterogeneous estimates of $\theta(x)$. As usual for M-estimator, we focus on expected score function and the equation which need to be satisfied is:

$$M_\theta, \nu(x) = E\left[\psi_{\theta(x),\nu(x)}\left(O_i\right)|X_i = x\right] = 0 \text{ for all } x \in \mathcal{X} \tag{16}$$

As it was discussed in section 2.3 the estimates, which are obtained by the method of moments are in general biased and therefore classical approach of taking an average of the estimators from different trees to obtain final prediction will not reduce the bias. Moreover, unlike in the classical approach GRF is not looking to conduct a split based on the general rule of improving the fit of the tree, but rather problem-specific splitting rules, focused on observing heterogeneity in the parameter of interest. The estimator obtained by the Generelized Randome Forest is asymptotically normally distributed and consistent, which makes the statistical inference quite straight forward.

In order to achieve this result, the goal of Athey's paper is to establish the Gaussianity of the $\hat{\theta}_{GRF}$. The asymptotic properties of the the $\hat{\theta}_{GRF}$ will be discuss in the last part of this section. However, in contrast to linear models, the nonlinear models require explicit assumptions on the parameter space for the asymptotic analysis (Wooldridge 2001).Therefore, the method assumes that both the covariates and parameters are in Euclidean space. Moreover, the GRF algorithm is build on the following assumptions:

1.$Lipschitz x - signal$: to ensure pointwaise convergence in x of the expected score function M (equation 29). This condition is a strong form of uniform continuity.

2.*Smooth identification* : As in the M-estimator, we assume that M-function is twice continues (it is possible to find second moment) with uniformly bounded second derivative and inevitable variance. We want to ensure uniqueness of the estimated parameter.

3.$Lipschitz(\theta, \nu)$-variogram : The score functions $\psi_{\theta(x),\nu(x)}\left(O_i\right)$ have continuous covariate structure.

4.*Regularity of* ψ : The ψ - function is Boreal measurable

5.*Existence of solution* For any weights α_i where their sum add up to 1 it is possible to find a solution which minimizes the moment condition.

6.*Concavity* The score function is a negative sub-gradient of the convex function. The solution of the equation 29 is a true minimum.

4.2 Weights

In order to avoid the problem of bias estimation of the single trees, Athey's develops a random forest weighting function, which measures the relevance of the $i-th$ observation to estimation of θ at x. The observations which end up in the same terminal node as our observation x are weighted evenly, where as the observations in other nodes receive the weight 0. The weight for one tree is calculated via $\alpha_{bi}(x) = \frac{\mathbf{1}(\{X_i \in L_b(x)\})}{|L_b(x)|}$, where leaves $L_b(x)$ with smaller number of elements receive higher weights (the tree was able to predict the estimate more accurately). To obtain the final weight for the $i-th$ observation, an average of all the weights across B trees is taken: $\alpha_i(x) = \frac{1}{B}\sum_{b=1}^{B}\alpha_{bi}(x)$. See 4 for visualization. The parameter is then estimated through minimization of the weighted sum of the moment function, which will allow us to obtain the weighted parameter as close to the sample moment:

$$(\hat{\theta}(x), \hat{\nu}(x)) \in \text{argmin}_{\theta,\nu} \left\{ \left\| \sum_{i=1}^{n} \alpha_i(x)\psi_{\theta,\nu}\left(O_i\right) \right\|_2 \right\} \tag{17}$$

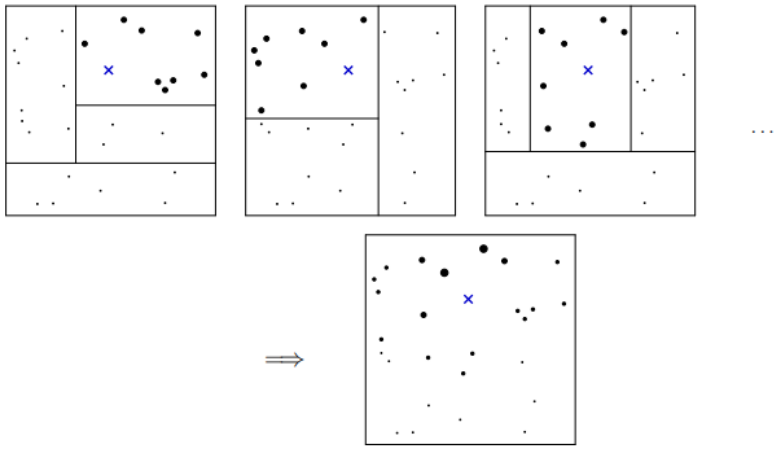

Figure 4: *Generalized random forest weighting function*
The weights are being assigned according to nearest neighbor rule where observations in the same leave are weighted equally and the final weights for the observations are calculated (Athey et al. 2019)

4.3 Split

Just like in the classical model, Athey's trees split the branches to immediately improve their fit. The purpose of the technique is to minimize the not weighted equation 17 for the available subset of data J. Just like in the classical approach, the split should minimize the error of prediction in each child node (equation 6). However, rather then focusing on the average outcome within the nodes R1 and R2, the method focuses on observing heretogenity in the parent node with observations $n_P = |\{i \in \mathcal{J} : X_i \in P\}|$ as quickly as possible. Therefore, the split will focus on finding children for which the parameter of interest θ will be different. Consequently, the split criteria is determined as following:

$$\Delta\left(C_1, C_2\right) := n_{C_1} n_{C_2} / n_P^2 \left(\hat{\theta}_{C_1}(\mathcal{J}) - \hat{\theta}_{C_2}(\mathcal{J})\right)^2 \tag{18}$$

The parameters θ_{C_2} and θ_{C_1} are the solutions of equation17 for the children nodes. Therefore if we assume that n_{C_1} n_{C_2} are fixed we can estimate the children prediction error as :

$$\text{err}\left(C_1, C_2\right) = K(P) - E\left[\Delta\left(C_1, C_2\right)\right] + o\left(r^2\right) \tag{19}$$

Where $K(P)$ is the measure of purity of the parent node (independent of how it is split) and $o\left(r^2\right)$ is taking into account the sampling variance and stabilizes the tree. As it can be seen from the equation, the higher purity of the parent node and the higher the $\Delta - criterion$ the smaller the error of the split. However, optimizing the criterion $\Delta - criterion$ through solving equation 17 for every possible combination proved to be computationally expensive. Therefore, a method of gradient-based approximation was considered. The method approximates the parameter $\tilde{\theta}_C \approx \hat{\theta}_C$ using two steps : labeling step and regression step. In the first step, a consistent estimate of the gradient (A_p) of the expectation of the moment function. As we want to take into account all the observation in the parent nod, we use the average of the sum of the gradients for each observation. The A_p is given as:

$$A_P = \frac{1}{|\{i : X_i \in P\}|} \sum_{\{i:X_i \in P\}} \nabla \psi_{\hat{\theta}_P, \hat{\nu}_P}(O_i) \tag{20}$$

The coefficient of the child node $(\tilde{\theta}_C)$ is estimated in the Newton Step fashion from the estimate of the parent node $(\hat{\theta}_P)$. The relation between the estimators is given by the equation:

$$\tilde{\theta}_C = \hat{\theta}_P - \frac{1}{|\{i : X_i \in C\}|} \sum_{\{i:X_i \in C\}} \xi^\top A_P^{-1} \psi_{\hat{\theta}_P, \hat{\nu}_P}(O_i) \tag{21}$$

The parameters of the parent node are obtained by solving equation 17 with vector ξ we chose the parameters of interest from the parameters vector. The second component of the right had side of the equation simply gives an average influence function of the $i - th$ observation which ends up in the children node in estimating the parameter in the parent node. We will use it to estimate the pseudo-outcomes in the children nodes:

$$\rho_i = -\xi^\top A_P^{-1} \psi_{\hat{\theta}_P, \hat{\nu}_P}(O_i) \in R \tag{22}$$

To understand this steps it helps to think of maximum likelihood estimation. Based on where the given $i - th$ observation is on the parameter distribution, we are able to predict what value of the predictor it takes. Obtaining the gradient matrix, the parent parameters and the pseudo-outcomes is considered the labeling step of the algorithm. The regression step uses the pseudo-outcomes to run the standard CART and obtains split which maximizes the $\Delta - criterion$ defined as:

$$\widetilde{\Delta}(C_1, C_2) = \sum_{j=1}^{2} \frac{1}{|\{i : X_i \in C_j\}|} \left(\sum_{\{:X_i \in C_j\}} \rho_i \right)^2 \tag{23}$$

After the split is conducted, the observations are relabeled by solving the local moment estimating equation and consequently obtaining the $(\hat{\theta}_C)$ for each node.

4.4 Building the forest

Based on the individual trees a forest is build. Each of the trees provides small noisy neighborhood relevant for x a a bias estimate of $(\theta_{GRF}(x))$. For the estimation of the stable forest-based $(\hat{\theta_{GRF}}(x))$ we create a bigger neighborhood for observation x based on the weighting system described in section 4.2. The the final estimate is obtained by solving equation 17. To obtain statistically valid results, Athey follows her honest forest approach by building the trees and estimating the parameters with different subsample of the training data as described in the section 3.3. For explanation of grf algorithms step by step please section 5.2.

4.5 Asymptotic Properties

To ensure consistency and statistical validity of the estimators both central limit theorem and the asymptotic variance needs to derived for the Generalized Random Forest. In the classical random forest U statistics can be used as regression forests are averages of regression trees for different subsamples. The U- statistics arise when estimating minimum- variance unbiased estimators. However, in the case of GRF we are not dealing with averages but weighted single moment equation. Therefore, a pseudo- random forest which can be evaluated with U-statistics needs to be estimated. After derivation of the pseudo-random forest and obtaining the pseudo-estimate from it Athey shows that pseudo and GRF estimator couples at the rate, which assures the consistency of the $\hat{\theta}_{GRF}$ estimator.

To estimate the pseudo-forest Athey utilizes the idea of influence function which is very strongly linked to the idea of the pseudo outcomes from the equation 32. We can denotes $\rho_i^*(x)$influence function as:

$$\rho_i^*(x) := -\xi^\top V(x)^{-1} \psi_{\theta(x),\nu(x)}\left(O_i\right) \tag{24}$$

Here, however we assume equation 24 to depend on true, unknown parameter $\theta(x)$. We can use the true parameter and the influence function at x to estimate the parameter of the pseudo-forest:

$$\tilde{\theta}^*(x) := \theta(x) + \sum_{i=1}^{n} \alpha_i(x) \rho_i^*(x) \tag{25}$$

We can see that the parameter of x can be estimated through a linear approximation of the true parameter plus the sum of weighted influence functions. Of course, this estimation is not feasible, however the pseudo-forest is U statistics and since it is a linear function of $\rho_i^*(x)$ we can write the final estimate as an average of pseudo-trees predictions $\tilde{\theta}^*(x) = \frac{1}{B}\sum_{b=1}^{B} \tilde{\theta}_b^*(x)$.
The biggest technical contribution of the Generalized Random Forest paper is the proof that the $\tilde{\theta}^*(x)$ and $\hat{\theta_{GRF}}(x)$ couple asymptotically at the rate which allows to focus on asymptotic properties of pseudo-forest. Moreover, the GRF estimator converges in probability to its true value. For proof and further technical discussion please refer to Athey et al. (2019) and Wager & Athey (2018).

5 Random vs Generalized Random Forest

5.1 Overview

Generalized random forest is a method for non-parametric estimation based on the approach of the random forest. The two methods have the same structure, however the aim as well as the steps conducted at every step of the algorithm vary significantly.
The objective of the random forest is accurate outcome prediction, whereas generalized random forest focuses on prediction of heterogeneous estimators.The moment condition, which is at the core of the estimation $E\left[\psi_{\theta(x),\nu(x)}\left(O_i\right)|X_i = x\right] = 0$ for all $x \in \mathcal{X}$ allow us to identify the conditional mean, quantiles, average partial effects ect. We are able to estimate any model from the M-estimator family. Knowing the moment condition allows for usage if wide range of statistical properties associated with it, whereas standard random forest is quite resistant to the statistical analysis.
Solving local moment conditions in generalized random forest gives us unfortunately biased estimators due to specificity of the moment estimators, even though we use the honest forest approach of splitting the sample for the partitioning and estimation of the parameter. Therefore averaging of the predictions would not make impact in terms of stabilizing of the prediction through reduction of the variance at it is achieved with the standard bagging approach. As it was presented in section 4, estimating the set of weighted set of neighbors for each test point x and then solving the cumulative weighted moment condition solves this issue. If the model which we are looking at aim to estimate conditional mean, the generalized random forest will perform equally good as the standard approach. To illustrate this point let's have a look at the simplest case of least-squares regression with moment condition:

$$\psi_{\theta(x)}(Y) = Y - \theta(x) \tag{26}$$

In the labeling step we get pseudo-outcome as the difference in the outcome and the average outcome in the parent nod ($\rho_i = Y_i - \overline{Y}_p$) and the maximizing step corresponds to the standard splitting approach. Therefore, in the generalized random forest the moment condition which defines the type of estimator one wants to obtain is at the core of the success of the method. The benefit of the generalized random forest can be seen in more with more complicated terms to be estimated.
Moreover, the adaptive neighborhood function obtained via partitioning needs to correctly capture the heterogeneity in the base function we try to estimate in order for the estimators to be consistent. Depending on the research question one try to tackle with the model, the generalized random forest will look quite different due to problem-specific splitting rules driven by different neighborhood functions. Whereas in the standard random forest we only care about prediction of one outcome, the splitting rules are universal for the regression. In both case the standard and the generalized random forest uses the random subset of the training data and random split selection to randomize the trees.

Over the years, there has been a vast development of random forest algorithms, which which aim at prediction of specific parameters such as *quantregForest* developed for quantile regression by Meinshausen (2006). However, as it will be shown in Section 6, *quantregForest* package perform still less good than the generalized random forest. The heterogeneity focused problem-specific split is much more sensitive to a change in the distribution of the observations with the leaf and not only its mean.

5.2 Algorithms Comparison

	Random Forest	**Generalized Ransom Forest**
Purpose	Prediction	Estimation of highly heterogeneous $\theta(x)$ parameters
Estimating Equation	$\hat{f}_{\text{rf}}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$	$E\left[\psi_{\theta(x),\nu(x)}\left(O_i\right)\left\|X_i = x\right.\right] = 0$ for all $x \in \mathcal{X}$
Assumptions	N/A	1.Lipschitz x-signal 2.Smooth Identification 3.Lipschitz(θ,ν)-variogram 4.$Regularity of \psi$ 5.Existence of solution 6.Concavity
	Algorithm	
Splitting the Data	Training, Testing Subset	Training, Estimating and Testing Subset
Building the Tree	1.Grow a tree by re-cursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached. i. Select m variables at random from the p variables. ii. Pick the best variable/split-point among the m. iii. Split the node into two daughter nodes	1.GradientTree(set of examples J , domain X) 2. node $P_0 \leftarrow CreateNode$(J,X) 3. queue Q $\leftarrow InitializeQueue(P_0)$ 4: while NotNull(node P $\leftarrow Pop(Q))do$ 5: $\left(\hat{\theta}_P, \hat{\nu}_P, A_P\right) \leftarrow SolveEstimatingEquation(P)$ 6: $R_P \leftarrow GetPseudoOutcomes\left(\hat{\theta}_P, \hat{\nu}_P, A_P\right)$ 7: split $\Sigma \leftarrow MakeCartSplit(P, RP)$ 8: if SplitSucceeded$(\Sigma)then$ 9: SetChildren (P, GetLeftChild $(\Sigma), GetRightChild(\Sigma)$) 10: AddToQueue(Q, GetLeftChild (Σ)) 11: AddToQueue(Q, GetRightChild (Σ)) 12: output tree with root node P_0
Building the Forest	1: For b = 1 to B : Draw a bootstrap sample Z of size N from the training data 2: Grow a random-forest tree Tb to the bootstrapped data. 3: Output the ensemble of trees $\{T_b\}_1^B$ To make a prediction at a new point x: Regression: $\hat{f}_{\text{rf}}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$ Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $C_{\text{rf}}^{B}(x) =$ majority vote $\{C_b(x)\}_1^B$	1. GeneralizedRandomForest(set of examples S, test point x) 2: weight vector $\leftarrow Zeros(\|S\|)$ 3: for b = 1 to total number of trees B do 4: set of examples $I \leftarrow Subsample(S, s)$ 5: sets of examples J_1, $J_2 \leftarrow SplitSample$(I) 6: tree T $\leftarrow GradientTree$(J_1, X) 7: N $\leftarrow Neighbors$(x,T,J_2) . Returns those elements of J_2 that fall into the same leaf as x in the tree T . 8: for all example $e \in Ndo$ 9: $\alpha[e] += 1/\|N\|$ 10: output $\theta(x)$, the solution to (2) with weights α/B
Split Equation	$\min_{j,s}\Big[\min_{c_1}\sum_{x_i\in R_1(j,s)}\left(y_i - c_1\right)^2 + \min_{c_2}\sum_{x_i\in R_2(j,s)}\left(y_i - c_2\right)^2$	$\widetilde{\Delta}\left(C_1, C_2\right) = \sum_{j=1}^{2}\frac{1}{\|\{i:X_i\in C_j\}\|}\left(\sum_{\{:X_i\in C_j\}}\rho_i\right)^2$

Table 2: *Random Forest vs GRF Algorithm*

6 Quantile Regression using *grf*

6.1 Quantile Regression

One of the applications of the *grf* presented in the paper is the quantile regression. To show the superiority of the generalized random model over the existing approach of Meinshausen (2006), who was the first one to develop the random forest algorithm for the quantile regression. Even though the estimation equation for both algorithms is essentially the same, the biggest difference is observed in the splitting rule, which in case of Meinshausen (2006) is not tailored to the quantile regression and therefore is only responsive the the change in the mean and not scale of the data distribution. The point is presented in figure 6. For the q-th quantile regression we use the moment function :

$$\psi_\theta (Y_i) = q\mathbf{1} (\{Y_i > \theta\}) - (1-q)\mathbf{1} (\{Y_i \leq \theta\}) \tag{27}$$

The pseudo-outcome, which is at the core of the splitting decision is derived as $\rho_i = \mathbf{1} (\{Y_i > \theta_{q,P} (X_i)\})$ where we categories the observations depending on whether is falls bellow or above the q-th quantify of the parent node.

As it can be seen in the Figure 6 both methods are able to pick up on the shift in the mean of the conditional quantile where x= 0. However, if the 0.5 quantile is unchanged and only 0.1 and 0.9 shift (scale shift) the approach suggested by Meinshausen (2006) is not able to pick it up and does not conduct the split since the split is conducted in the same way as the regression splits.Using Athey's approach it is also possible to estimate number of quantiles at the same time and use multiples classification rule that splits the observations into intervals.

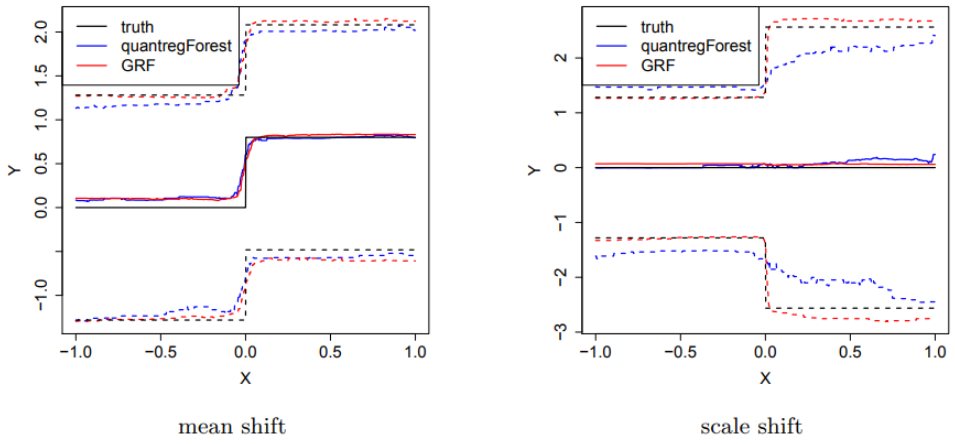

Figure 5: *Comparison of quantile regression using generalized random forests and the quantregForest package of Meinshausen (2006)*
(Athey et al. 2019) Figure 2

The paper showcases the superiority of its results in comparison to other random forest algorithms. However, it lacks comparison to other machine learning techniques that could be used for the hetergoenic parameters estimation. Some data science block such as *TowardsDataScience* suggest that deep learning techniques for quantile regression can be another variable approach for estimation , however the gain in terms of total quantile loss is not so significant when we deal with quite complex model. Therefore, it seems that with data sets of high dimensions it is a good idea to use machine learning rather than OLS or standard quantile regression.

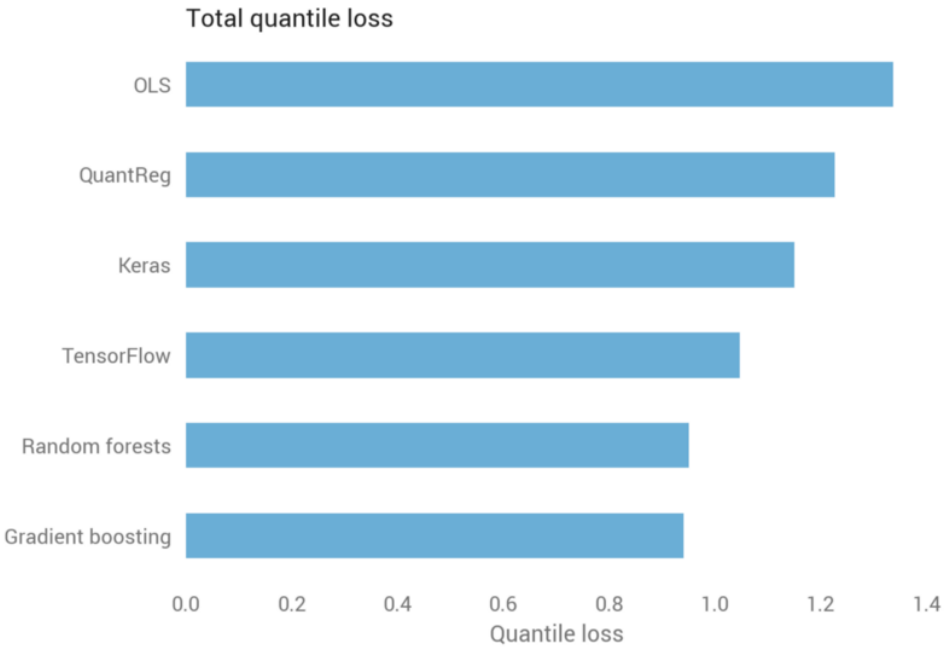

Figure 6: *Comparison of methods for approximation of the quantile regression* Large sample estimation methods vs their accuracy. Source: Towards data science

7 Conclusion

The generalized random forest allows for adaptation of random forest technique to the purpose of heterogeneous parameters prediction. The model has used the generalized method of moments to solve the sample equation, which by the analog principle, also holds true for the population.The generalized random forest has proven to perform better than the other random forest algorithms for the causal heterogeneity estimation.

The generelized random forest has proven to perform better than the other random forest algorithms. The better results have been attributed to the problem specific split function which conducts the split of the data not only according to their mean value, but also according the specific nature of the parameter the model tries to estimate. Another advantage is that the asymptotic properties of the generelized random forest allow us to conduct the statistical inference of the estimated coefficients.

The method struggles with the same limitation as any other non-parametric, which the authors mention themselves, is the fact that the confidence intervals depend on undesmoothing (we make sure that we can also account for the bias of the forest), which makes them not quite accurate. The next step of the algorithm development would be to develop confidence intervals, which are bias-correct. The other difficulties arise from the assumption of the parameter space, which can create not stable results when we approach the edge of the compact space. However, we knows those limitation hold for any maximum likelihood estimations.

The model performs well in the models with high dimensions. However, the question as always arises about the applicability of the method for the econometric purposes due to limited size of the samples. One could argue that the adaptation should work in the opposite direction where econometric methods are applied for the purpose of the machine learning, where for example personalized recommendations should be considered causal problem and not prediction problem as it has been seen up till now.

Handout: Generalized Random Forest

Author: Malgorzata Olesiewicz, Paper : Generalized Random Forest (Athey et al. 2019)

Summary
The generalized random forest allows for adaptation of random forest technique to the purpose of heterogeneous parameters prediction. The model has used the generalized method of moments to solve the sample equation, which by the analog principle, also holds true for the population.The generalized random forest has proven to perform better than the other random forest algorithms for the causal heterogeneity estimation. The better results have been attributed to the problem specific split function, which conducts the split of the data according to the specific nature of the parameter one tries to estimate. The model performs well in high dimensional setting and could be used for personalized recommendations, which should be considered as a causal and not prediction problem.

Random Forest - Survival Kit
Classification and Regression Tree (CART) Decision tree classifies data according to the observations'characteristics. The underling idea here is that observations with similar characteristics will most likely have similar outcomes. The tree is developed though recursive partitioning where observations are classified into subsamples, each of them with different outcome which can be a category or a constant .
Bagging (bootstrap aggregation) focuses on the redaction of variance of the prediction through randomization and averaging the predictions of many unbiased but noisy models.
U-Statistics is a commonly accepted notions of unbiased estimation such as the sample mean and the unbiased sample variance (the "U" stands for "unbiased") in a large sample models.
Honest Tree The ultimate goal of the "honest" CART's algoritms $\pi(.)$ used on the training sample $S^{te}$is to maximize the honest criterion:

$$Q^H(\pi) \equiv -E_{\mathcal{S}^{\text{ est}},\mathcal{S}^{\text{ est}},\mathcal{S}^{\text{ est}}}\left[\text{MSE}\left(\mathcal{S}^{\text{ te}},\mathcal{S}^{\text{ est}},\pi\left(\mathcal{S}^{\text{tr}}\right)\right)\right] \tag{28}$$

where S^{te} stand for the test sample and S^{est} for the estimation sample. The estimated coefficients in each leaf is unbiased as the sample used for the partitioning is different than for its estimation.

Key equations
Estimation Equation

$$M_\theta, \nu(x) = E\left[\psi_{\theta(x),\nu(x)}\left(O_i\right)|X_i = x\right] = 0 \text{ for all } x \in \mathcal{X} \tag{29}$$

Data: $(X_i, O_i) \in \mathcal{X} \times \mathcal{O}$, $\psi(\cdot)$: some scoring function, $\theta(x)$: parameter of interest, $\nu(x)$: optional nuisance.
The gradient (A_p) of the expectation of the moment function is given as:

$$A_P = \frac{1}{|\{i : X_i \in P\}|}\sum_{\{i:X_i\in P\}} \nabla\psi_{\hat{\theta}_P,\hat{\nu}_P}\left(O_i\right) \tag{30}$$

The coefficient of the child node $(\tilde{\theta}_C)$ is estimated in the Newton Step fashion from the estimate of the parent node $(\hat{\theta}_P)$. The relation between the estimators is given by the equation:

$$\tilde{\theta}_C = \hat{\theta}_P - \frac{1}{|\{i : X_i \in C\}|}\sum_{\{i:X_i\in C\}} \xi^\top A_P^{-1}\psi_{\hat{\theta}_P,\hat{\nu}_P}\left(O_i\right) \tag{31}$$

With vector ξ we chose the parameters of interest from the parameters vector. The second component of the right had side of the equation simply gives an average influence function of the $i-th$ observation which ends up in the children node in estimating the parameter in the parent node.
Pseudo-outcomes in the children nodes:

$$\rho_i = -\xi^\top A_P^{-1}\psi_{\hat{\theta}_P,\hat{\nu}_P}\left(O_i\right) \in R \tag{32}$$

The regression step uses the pseudo-outcomes to run the standard CART and obtains split which maximizes the $\Delta - criterion$ defined as:

$$\widetilde{\Delta}\left(C_1, C_2\right) = \sum_{j=1}^{2}\frac{1}{|\{i : X_i \in C_j\}|}\left(\sum_{\{:X_i\in C_j\}}\rho_i\right)^2 \tag{33}$$

	Random Forest	**Generalized Ransom Forest**
Purpose	Prediction	Estimation of highly heterogeneous $\theta(x)$ parameters
Estimating Equation	$\hat{f}_{\text{rf}}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$	$E\left[\psi_{\theta(x),\nu(x)}\left(O_i\right)\left\|X_i = x\right.\right] = 0$ for all $x \in \mathcal{X}$
Assumptions	N/A	1.Lipschitz x-signal 2.Smooth Identification 3.Lipschitz(θ,ν)-variogram 4.$Regularity of \psi$ 5.Existence of solution 6.Concavity
	Algorithm	
Splitting the Data	Training, Testing Subset	Training, Estimating and Testing Subset
Building the Tree	1.Grow a tree by re-cursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached. i. Select m variables at random from the p variables. ii. Pick the best variable/split-point among the m. iii. Split the node into two daughter nodes	1.GradientTree(set of examples J , domain X) 2. node $P_0 \leftarrow CreateNode$(J,X) 3. queue Q $\leftarrow InitializeQueue(P_0)$ 4: while NotNull(node P $\leftarrow Pop(Q))do$ 5: $\left(\hat{\theta}_P, \hat{\nu}_P, A_P\right) \leftarrow SolveEstimatingEquation(P)$ 6: $R_P \leftarrow GetPseudoOutcomes\left(\hat{\theta}_P, \hat{\nu}_P, A_P\right)$ 7: split $\Sigma \leftarrow MakeCartSplit(P, RP)$ 8: if SplitSucceeded$(\Sigma)then$ 9: SetChildren (P, GetLeftChild $(\Sigma), GetRightChild(\Sigma)$) 10: AddToQueue(Q, GetLeftChild (Σ)) 11: AddToQueue(Q, GetRightChild (Σ)) 12: output tree with root node P_0
Building the Forest	1: For b = 1 to B : Draw a bootstrap sample Z of size N from the training data 2: Grow a random-forest tree Tb to the bootstrapped data. 3: Output the ensemble of trees $\{T_b\}_1^B$ To make a prediction at a new point x: Regression: $\hat{f}_{\text{rf}}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$ Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $C_{\text{rf}}^{B}(x) =$ majority vote $\{C_b(x)\}_1^B$	1. GeneralizedRandomForest(set of examples S, test point x) 2: weight vector $\leftarrow Zeros(\|S\|)$ 3: for b = 1 to total number of trees B do 4: set of examples $I \leftarrow Subsample(S, s)$ 5: sets of examples J_1, $J_2 \leftarrow SplitSample$(I) 6: tree T $\leftarrow GradientTree(J_1, X)$ 7: N $\leftarrow Neighbors$(x,T,J_2) . Returns those elements of J_2 that fall into the same leaf as x in the tree T . 8: for all example $e \in Ndo$ 9: $\alpha[e] += 1/\|N\|$ 10: output $\theta(x)$, the solution to (2) with weights α/B
Split Equation	$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}\left(y_i - c_1\right)^2 + \min_{c_2}\sum_{x_i\in R_2(j,s)}\left(y_i - c_2\right)^2\right.$	$\widetilde{\Delta}\left(C_1, C_2\right) = \sum_{j=1}^{2}\frac{1}{\|\{i:X_i\in C_j\}\|}\left(\sum_{\{:X_i\in C_j\}}\rho_i\right)^2$

References

Athey, S. & Imbens, G. (2016), 'Recursive partitioning for heterogeneous causal effects', *Proceedings of the National Academy of Sciences* **113**(27), 7353–7360.

Athey, S., Tibshirani, J., Wager, S. et al. (2019), 'Generalized random forests', *The Annals of Statistics* **47**(2), 1148–1178.

Hastie, T., Tibshirani, R. & Friedman, J. (2001), *The Elements of Statistical Learning*, Springer Series in Statistics, Springer New York Inc., New York, NY, USA.

Leeb, H. & Pötscher, B. M. (2008), 'Can one estimate the unconditional distribution of post-model-selection estimators?', *Econometric Theory* **24**(2), 338–376.

Meinshausen, N. (2006), 'Quantile regression forests', *Journal of Machine Learning Research* **7**(Jun), 983–999.

Wager, S. & Athey, S. (2018), 'Estimation and inference of heterogeneous treatment effects using random forests', *Journal of the American Statistical Association* **113**(523), 1228–1242.

Wooldridge, J. M. (2001), *Econometric Analysis of Cross Section and Panel Data*, Vol. 1, The MIT Press.

Yuan, A., Giurcanu, M., Luta, G. & Tan, M. T. (2017), 'U-statistics with conditional kernels for incomplete data models', *Annals of the Institute of Statistical Mathematics* **69**(2), 271–302.