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	$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}\left(y_i - c_1\right)^2 + \min_{c_2}\sum_{x_i\in R_2(j,s)}\left(y_i - c_2\right)^2\right.$	$\widetilde{\Delta}\left(C_1, C_2\right) = \sum_{j=1}^{2}\frac{1}{\|\{i:X_i\in C_j\}\|}\left(\sum_{\{:X_i\in C_j\}}\rho_i\right)^2$