Generalized random forests, an introduction

Project for the course MATH-412, EPFL

William Cappelletti, MA, Charles Dufour, MA

December 2018

Abstract

The Generalized random forests method was developed for non-parametric statistical estimation and is based on Random Forests [4]. It can be used to fit any quantity which is implicitly defined as the solution to a local moment equation. To solve it, we use an empirical estimate given by a weighted sum of nearby training examples, which weights are obtained through a forest of problem-specific trees. We discuss how to build these data-adaptive trees and we give an application example.

1 Introduction

Random forests are widely used in machine learning nowadays due to their intuitive approach and explicable process. They were developed by Breiman [4] in 2001 and they are generally used for non-parametric estimation of conditional mean $\mathbb{E}[Y|X]$.

S. Athey, J. Tibshirani and S. Wager, in their paper *Generalized Random Forests* [1], extend this concept for the estimation of any quantity $\theta(X)$ which can be defined via a moment equation. More precisely, given n independent data points $(X_i, O_i) \in \mathcal{X} \times \mathcal{O}$, we want to find $\theta(x)$ solving the local moment equation:

$$\mathbb{E}\left[\psi_{\theta(x),\nu(x)}(O_i)|X_i = x\right] = 0 \quad \forall x \in \mathcal{X}$$

We give a brief overview on the reasons why it is an interesting approach and on the main differences from Breiman's algorithm. Then, we discuss the computational part and the issues arisen in practice by the construction of problem-specific trees. Asymptotic results, for instance normality and consistency of the estimators, are cited, but not proven (all proof are in the annexes of *Generalized Random Forests* by S. Athey, J. Tibshirani and S. Wager [1]).

2 CART and random forests

Trees alone are bad predictors, since their predictions are subject to high variance, but they are computationally cheap and they are easily explained. Here we quickly explain how trees are usually built, and how we can combine them in better estimators, the random forests (see [4] for more details).

In Breiman's forests [4], trees use axis aligned splits to divide the feature space in areas onto which it fits a piece-wise constant estimator of the response. While growing each tree, those splits are chosen greedily at each step to get the biggest change in goodness of fit, quantified by misclassification error, Gini index, Cross-entropy or some hypothetical loss function. The approach presented in [4] is called CART, and CART regression splits are only sensitive to changes in the conditional mean of O given X [1].

Many trees are grown on subsets of the data, with or without bootstrapping, and in the end the prediction results are averaged. This approach gives nice results in the estimation of the conditional mean $\mathbb{E}[Y|X]$ since it drastically reduces the variance of each tree, while not increasing the bias too much; nevertheless, it is limited on the hypothesis and does not account for underlying heterogeneity and specificity of the dataset.

3 Generalized random forests (GRF)

Extending this approach, we aim to find a flexible method for estimating any quantity $\theta(x)$ identified implicitly by a local moment equation, and not only by the conditional mean.

3.1 Solution to local estimating equations

The main idea in GRF is, given n independent data points $(X_i, O_i) \in \mathcal{X} \times \mathcal{O}$, to seek estimates of $\theta(x)$ defined by the following local equation:

$$\mathbb{E}\left[\psi_{\theta(x),\nu(x)}(O_i)|X_i = x\right] = 0 \quad \forall x \in \mathcal{X} \tag{1}$$

where $\psi(.)$ is a score function and $\nu(.)$ is an optional nuisance parameter. We can see the maximum likelihood as a special case of our setting, supposing $O_i|X_i = x \sim f_{\theta(x),\nu(x)}$. Then, solving the estimating equation (1) using

$$\psi_{\theta(x),\nu(x)} = \nabla log\left(f_{\theta(x),\nu(x)}(O_i)\right)$$

gives the local maximum likelihood parameters $(\hat{\theta}(x), \hat{\nu}(x))$.

To solve the equation (1), we use the following empirical estimate [1]:

$$\left(\hat{\theta}(x), \hat{\nu}(x)\right) \in \text{argmin}_{\theta,\nu}\left\{\left\|\sum_{i=1}^{n} \alpha_i(x)\psi_{\theta,\nu}(O_i)\right\|_2\right\}. \tag{2}$$

Here, the α_i represent the relevance of the i^{th} training set to fitting $\widehat{\theta}(x)$. Since the original equation (1) is conditioned on $X = x_i$, it seems reasonable tha,t when we try to approximate it with our data, we give more importance to the observation (X_0, O_0) with X_0 "near" x_i, where "near" depends on a metric still to be defined.

The weights are usually computed using deterministic kernel functions, which are subject to the curse of dimensionality. GRF goal is to use forests, instead, to find adaptive weights depending on the target function. These weights are computed, as shown in Algo 1, by growing problem-specific trees as described in section 4 and they define a forest based adaptive neighbourhood of x, capturing the frequency with which the i-th training example falls into the same leaf as x.

Algorithm 1: Computation of the weights $\alpha_i(x)$

```
  Result: Weights $\alpha_i(x)$
1 Fix the number $B$ of trees;
2 for $b$ = 1 to $B$ do
3     Randomly split the data point in two evenly-sized, non-overlapping halves $\mathcal{J}_1, \mathcal{J}_2$;
4     Fully grow an unpruned tree $T_b$ using $\mathcal{J}_1$;
5     Define the set $L_b(x)$ as the set of observations of $\mathcal{J}_2$ that fall in the same leaf as $x$ in $T_b$ ;
6     Compute for each i such that $X_i \in \mathcal{J}_2$:
7         $\alpha_{bi}(x) = \mathbf{1}\left(\{X_i \in L_b(x)\}\right) / |L_b(x)|$;
8 end
  Output: $\alpha_i(x) = \frac{1}{B} \sum_{b=1}^{B} \alpha_{bi}(x)$
```

The process we use to split the data to grow the tree with one part and then repopulate it with the other part (line 5 in Algo 1) is called *honesty* (see Wager and Athey, 2018 [6]) for more details).

4 Gradient tree algorithm

4.1 Splitting criterion

In order to get a better approximation of (1), we need weights that capture the heterogeneity in the target functional $\theta(x)$, therefore they cannot be computed in the same way for all problems. GRF method focus on growing the trees in the forest using a criteria suitable for the specific problem. At first, the search of good splits proceeds greedily, like in Breiman's forests [4]. Therefore, the goal is to get splits of current nodes that rapidly improve the quality of fit as much as possible.

From a parent node $P \subseteq \mathcal{X}$, given a sample of data $\mathcal{J}$, we define $(\hat{\theta}_P, \hat{\nu}_P)(\mathcal{J})$ as the solution, in the node, of the estimating equation:

$$\left(\hat{\theta}_P, \hat{\nu}_P\right) \in \operatorname{argmin}_{\theta,\nu} \left\{ \left\| \sum_{i \in \mathcal{J} : X_i \in P} \psi_{\theta,\nu}(O_i) \right\|_2 \right\}. \tag{3}$$

Then, using an axis-aligned cut, we want to split P in two children C_1 and C_2 in which the θ-estimates are as accurate as possible. This translates in the minimization of the following error

$$\operatorname{err}(C_1, C_2) = \sum_{j=1,2} \mathbb{P}\left[X \in C_j | X \in P\right] \mathbb{E}\left[\left(\hat{\theta}_{C_j}(X) - \theta(X)\right)^2 \,\middle|\, X \in C_j \right], \tag{4}$$

where expectations are over the randomness of the fit in the child C_j, $\hat{\theta}_{C_j}$, and a new test point X. This is the usual L2 error, but in our setting, $\theta(X)$ is only accessible through the moment equation (1). Therefore, we do not have access to a good estimate of $\operatorname{err}(C_1, C_2)$.

We introduce a new quantity

$$\Delta(C_1, C_2) = \frac{n_1 n_2}{n_P^2} \left(\hat{\theta}_{C_1} - \hat{\theta}_{C_2}\right)^2, \tag{5}$$

in which n_1, n_2 and n_P are the number of observations in the two children and the parent nodes respectively. This Δ is an indicator of the heterogeneity of the children, as the term $(\hat{\theta}_{C_1} - \hat{\theta}_{C_2})^2$

increases as the estimates in the children get further apart and the ratio $n_1 n_2 / n_p^2$ is maximized when the children contains the same number of observations.

Instead of minimizing the error $\mathrm{err}(C_1, C_2)$, we can maximize the children heterogeneity $\Delta(C_1, C_2)$. This holds since, under some regularity conditions described in subsection 5.1, we can write the error as

$$\mathrm{err}(C_1, C_2) = K(P) - \mathbb{E}[\Delta(C_1, C_2)] + o(r^2), \tag{6}$$

where $n_{C_1}, n_{C-2} \gg r^{-2}$. Here r is defined as the radius of the node, and $K(P)$ is a term that only depends on the parent node, being a deterministic measure of the purity of P. The o incorporates terms that depend on the sampling variance of regression trees. At this point, we have a criterion that does not use the true parameter and we would like to perform splits to maximize it.

4.2 Gradient based approximations

Since $\hat{\theta}_{C_1}$ and $\hat{\theta}_{C_2}$ are solutions to (3), it is computationally infeasible to explicitly optimize them over all possible splits. Instead, we use an approximate $\tilde{\Delta}(C_1, C_2)$ using gradient-based approximations for $\hat{\theta}_{C_1}$ and $\hat{\theta}_{C_2}$.

The idea is to use one step of Newton-Raphson method, which is used to find the zero of a function, to approximate, knowing the parent estimate, the minimizers of (3) in the children. In particular, we search for zeroes of $\mathbb{E}[\psi_{\hat{\theta},\hat{\nu}}(O_i)|X_i \in C_{1,2}]$, which would give approximate solutions to the estimating equation in the children. We recall that, given a point x_0 and a function f, one step of Newton-Raphson gives an approximate solution x_* to $f(x) = 0$ by taking

$$x_* = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

In our setting, we take $f(\hat{\theta}, \hat{\nu}) = \mathbb{E}[\psi_{\hat{\theta},\hat{\nu}}(O_i)|X_i \in P]$ and $x_0 = (\hat{\theta}_P, \hat{\nu}_P)$. Therefore, with A_P any consistent estimate of $\nabla \mathbb{E}[\psi_{\hat{\theta}_P,\hat{\nu}_P}(O_i)|X_i \in P]$, we can approximate $\hat{\theta}_C$ by

$$\tilde{\theta}_C = \hat{\theta}_P - \frac{1}{|\{i : X_i \in C\}|} \sum_{i:X_i \in C} \xi^T A_p^{-1} \psi_{\hat{\theta}_P,\hat{\nu}_P}(O_i), \tag{7}$$

where ξ is the unitary vector that isolate θ in (θ, ν). This intuitively holds considering that, if the number of observations in the child $n_c = |\{i : X_i \in C\}|$ goes to infinity, than the right hand term of (7) converges, by the law of large numbers, to $\hat{\theta}_P - \mathbb{E}[\xi^T A_P^{-1} \psi_{\hat{\theta},\hat{\nu}}(O_i)|X_i \in C]$. By consistency of A_P, which is deterministic once we condition on $X_i \in C \subset P$, this term is exactly

$$\hat{\theta}_P - \xi^T (\nabla f(\hat{\theta}, \hat{\nu}))^{-1} \mathbb{E}[\psi_{\hat{\theta}_P,\hat{\nu}_P}(O_i)|X_i \in C],$$

which is, indeed, a step of Newton-Raphson.

This reduces drastically the complexity of the problem, since, instead of solving (3) for each possible split, we just have to compute, for each i such that $X_i \in P$, $\rho_i = -\xi^T A_p^{-1} \psi_{\hat{\theta}_P,\hat{\nu}_P}(O_i)$ once in the parent and then we can use the ρ_i to compute the $\tilde{\Delta}(C_1, C_2)$, which can be rewritten as

$$\tilde{\Delta}(C_1, C_2) = \frac{n_1 n_2}{n_P^2} \left(\tilde{\theta}_{C_1} - \tilde{\theta}_{C_2} \right)^2 = \sum_{j=1}^{2} \frac{1}{n_j} \left(\sum_{i:X_i \in C_j} \rho_i \right)^2. \tag{8}$$

We can see the ρ_i as influence functions of the i-th observation to compute $\hat{\theta}_P$ in the parent, because of their gradient component. The split can therefore be considered as way of distributing the observations among the children in order to have the most overall influence on both sides (which is another way to interpret the heterogeneity).

It can be shown that, under the same assumptions of (6), $\Delta(C_1, C_2)$ and $\tilde{\Delta}(C_1, C_2)$ are approximately equivalent, in that $\tilde{\Delta}(C_1, C_2) = \Delta(C_1, C_2) + o_P(\max\{r^2, 1/n_1, 1/n_2\})$. Therefore, using this approximation, we have a computationally efficient way to find our splits.

4.3 Growing the trees

Thanks to the criteria introduced in previous sections, we are now able to grow trees which are specific to our problem. More specifically, we perform a recursive partitioning of the nodes, until we have only one element. At each node P, we can separate the procedure in two parts: a labelling step and a regression step. In the **labelling step**, we compute $\hat{\theta}_P, \hat{\nu}_P, A_P^{-1}$ and we use them to get pseudo outcomes $\rho_i = -\xi^T A_p^{-1} \psi_{\hat{\theta}_P, \hat{\nu}_P}(O_i)$ through one run over the node observations. Then, in the **regression step**,we split P into two axis-aligned children C_1 and C_2 such as to maximize the $\tilde{\Delta}(C_1, C_2)$ criterion given by (8).

It is now interesting to notice that it is in the labelling step that the specificity of the problem is encoded, since, once we get the pseudo-outcomes, the scanning performed in the regression step is shared across all different type of forests. Furthermore, when growing a tree, the computation is usually dominated by the split-selection step and in our setting, thanks to the fact that the criterion (8) is defined by cumulative sums, it is possible to easily evaluate all possible split points with only a single pass over the data along a given feature.

Finally, we now have an efficient and problem-specific way to compute the trees that we use in Algo 1.

5 Asymptotic results

5.1 Assumptions

Assumption 1 (Lipschitz x-signal). For fixed values of (θ, ν), we assume that $M_{\theta,\nu}(x) := \mathbb{E}\left[\psi_{\theta,\nu}(O)|X = x\right]$ is Lipschitz continuous in x.

Assumption 2 (Smooth identification). When x is fixed, $M_{\theta,\nu}(x)$ is twice differentiable in (θ, ν), with a uniformly bounded second derivative. $V(x) := V_{\theta(x),\nu(x)}(x)$ is invertible for all $x \in \mathcal{X}$, with

$$V_{\theta(x),\nu(x)}(x) = \frac{\partial}{\partial(\theta, \nu)} M_{\theta,\nu}(x)\bigg|_{\theta(x),\nu(x)}$$

Assumption 3 (Lipschitz (θ, ν)-variogram). The score functions $\psi_{\theta,\nu}(O_i)$ have a continuous covariance structure. Writing $\|\cdot\|_F$ for the Frobenius norm, we define γ to be the worst case variogram, i.e:

$$\gamma\left(\begin{pmatrix}\theta\\ \nu\end{pmatrix}, \begin{pmatrix}\theta'\\ \nu'\end{pmatrix}\right) := \sup_{x\in\mathcal{X}} \left\{\|\psi_{\theta,\nu}(O_i) - \psi_{\theta',\nu'}(O_i)|X_i = x\|_F\right\}$$

Then for some $L > 0$:

$$\gamma\left(\begin{pmatrix}\theta\\ \nu\end{pmatrix}, \begin{pmatrix}\theta'\\ \nu'\end{pmatrix}\right) \leq L \left\|\begin{pmatrix}\theta\\ \nu\end{pmatrix} - \begin{pmatrix}\theta'\\ \nu'\end{pmatrix}\right\|_2 \quad \text{for all } (\theta, \nu), (\theta', \nu') \tag{9}$$

Assumption 4 (Regularity of ψ).

$$\psi_{\theta,\nu}(O) = \lambda(\theta, \nu; O_i) + \zeta_{\theta,\nu}(g(O_i)) \tag{10}$$

With λ Lipschitz-continuous in (θ, ν), $g : \{O_i\} \mapsto \mathbb{R}$ is an univariate summary of O_i, and $\zeta_{\theta,\nu} : \mathbb{R} \mapsto \mathbb{R}$ is any family of monotone and bounded functions.

Assumption 5 (Existence of solutions). We assume that for any weights $\{\alpha_i\}_{i\in I}$ such that $\sum_i \alpha_i = 1$, solving equation (2) gives estimate $(\hat{\theta}, \hat{\nu})$ that at least solve partially the estimating equation, i.e.

$$\left\| \sum_{i=1}^{n} \alpha_i \psi_{(\hat{\theta},\hat{\nu})}(O_i) \right\|_2 \leq C \max\{\alpha_i\}$$

for some $C > 0$.

Assumption 6 (Convexity). The score function $\phi_{\theta,\nu}(O_i)$ is a negative subgradient of a convex function, and the expected score $M_{\theta,\nu}(X_i)$ is the negative gradient of a strongly convex function.

5.2 Asymptotic results

Under mild conditions on the structure of the trees and the honesty paradigm (see Algo 1), and with the assumptions 1-6 of the previous subsection, it is possible to show the following theorem (see [1] for the proofs).

Theorem 1 *The estimates $\hat{\theta}(x)$ based on the weights obtained from trees grown as described in subsection 4.1 are asymptotically consistent and Gaussian.*

6 Application: Quantile Regression Forests

We now compare the generalized random forest method with a standard random forests procedure, reproducing on application of the original paper [1].

We simulate the following data $(X_i, O_i) \in [-1; 1]^{40}$, with $X_i \sim \text{Unif}[-1, 1]$, and $Y_i|(X_i)_1 \sim \mathcal{N}\left(0, (1 + \mathbf{1}\{(X_i)_1 > 0\})^2\right)$, while the others 39 covariates are just noise. We then estimate the quantiles at 0.1,0.5,0.9.

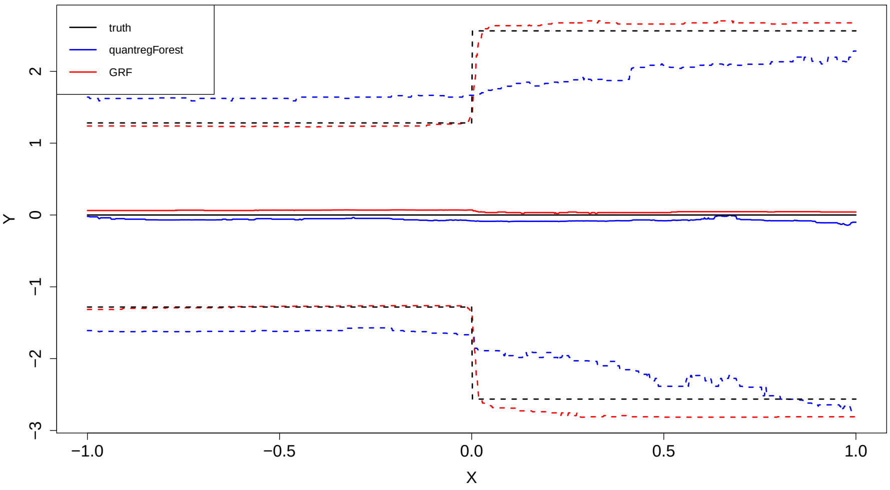

Figure 1: Comparison between the `grf` and `quantregForest` packages, $2'000$ data points simulated

In our setting, the q-th quantile is identified via (1) using $\psi_\theta(Y_i) = q\mathbf{1}_{\{Y_i>\theta\}} - (1-q)\mathbf{1}_{\{Y_i<\theta\}}$ and we compare the GRF approach to the approach developed by Meinshausen (2006) [5], a forest based quantile regression algorithm. This approach still fits the paradigm to solve a moment equation using weights, but these weights are derived from trees using usual regression splits, following the CART rules.

We can see in Fig. 1 that the usual random forests has trouble adjusting to the change in the variance, while the generalized random forests reproduces it quite accurately. This is due to the fact that GRF targets specifically the functional we want to approximate, while the other method only is only sensitive to changes in the conditional mean.

7 Conclusion

The GRF methods introduces a much more general framework than the usual random forest and can be used in a wide range of statistical models where usual random forests can not be applied or are not as efficient. Here we only present one example of GRF application, namely on quantile regression, but more can be found in the original paper [1].

The software is easy to use and modular. The labelling step, which is the most specific part, is coded efficiently and is easily reusable to fit any score function. Therefore, one can use the software without having to worry too much about performances.

A more in depth and interesting approach would be to study the proofs and look at comparison between the running time and accuracy of this method when compared to others machine learning models.

References

[1] Susan Athey, Julie Tibshirani and Stefan Wager, 2016. *Generalized Random Forests*, Papers 1610.01271, arXiv.org, revised Apr 2018

[2] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning.* New York, NY, USA: Springer New York Inc., 2001.

[3] Fan, J. and Farmen, M. and Gijbels, I., *Local maximum likelihood estimation and inference*, Journal of the Royal Statistical Society: Series B (Statistical Methodology). Wiley Online Library, 1998.

[4] Breiman, L., Friedman, J., Stone, C.J. and Olshen, R.A. (1984). *Classification and Regression Trees.* CRC press

[5] Meinshausen, N. (2006). *Quantile regression forests.* JMLR 7 983–999.

[6] Wager, S. and Athey, S. (2018). *Estimation and inference of heterogeneous treatment effects using random forests.* **JASA** just-accepted