Subgroup Discovery to Select Decision Trees from a Random Forest

Course: 2019-201300074-2B, Group: 15, Submission Date: June 28, 2020

Sathvik Guru Rao
University of Twente
s.gururao@student.utwente.nl

Nikita Meier
University of Twente
n.meier-1@student.utwente.nl

ABSTRACT

Random forest usually seems to be more accurate model than decision tree, but the major drawback of random forest model is the lack of interpretability, while decision trees are transparent and may be interpreted in a simple way. In this paper subgroup discovery method is applied to random forest in order to select subgroups of entities in the dataset, which seems to be able to approximate with a single random tree. This provides us with the interpretable subgroups of dataset, which makes the random forest model less black-box than usual.

KEYWORDS

Decision tree, random forest, bagging, subgroup discovery, exceptional model mining.

1 INTRODUCTION

One of the essential topics in data mining is interpretability of the model predictions. For many existing machine learning techniques, results may seem unexplainable in a sense that we cannot see clearly how these predictions were obtained from the given data, making us not being able to explain the reasons standing after such predictions. Unexplainable models are often called black-box, because one cannot clearly see what exactly is going on inside. On opposite to black-box models, some machine learning techniques, such as decision trees and linear regression, may be easily interpreted based on given data.

Random forest, which is basically the bagging of decision tree ensemble, seems to be an unexplainable model. The main task of this paper is to modify the random forest model in order to make it more interpretable. The way to achieve this is changing random forest predictions on some subgroups of the data to decision tree predictions, which leads to some lack in accuracy, but substantially increases the interpretability of the model on such subgroups. Also, the second goal of this paper is to explore the trade-off between accuracy and interpretability of such built model.

The paper is organised as follows. Subsections 1.1, 1.2 recaps the basic concepts used for this paper, Section 2 gives an overview of existing methods for enhancing explainability, Section 3 describes overall methodology of work, Section 4 provides results, Section 5 is conclusion and comments on results and Section **??** describes contributions of each author.

1.1 Random forest

The random forest method is suitable for both classification and regression.It is based on constructing several decision trees and then aggregating the predictions which usually leads to a model with better accuracy, compared to plain decision tree algorithm. The idea of random forest is to improve the variance reduction of bagging by reducing the correlation between the trees without increasing the variance too much. When used for classification, a random forest obtains a class vote from each tree, and then classifies using majority vote. When used for regression, the predictions from each tree are simply averaged. The reduction in variance stabilises the model, reduces its bias towards choices of training data, and leads to less variable and more accurate predictions.

1.2 Subgroup discovery

Subgroup discovery is the technique used to extract some meaningful rules with respect to the target variable. Subgroup may be understood as some pattern extracted in the form of rules. As input for subgroup discovery we are given observations and their properties. The task of subgroup discovery is to discover the subgroups of the observations that are statistically different from average in sense of target variable. Diferent measures may be used to quantify the interestingness of such subgroups.

2 RELATED WORK

Different model interpretation techniques may be split in two big groups: model-agnostic and model-specific solutions.

As it follows from the name, model-agnostic techniques are indifferent for the model choice, making them applicable to any prediction model. One of the most well-known basic model-agnostic techniques for interpretation is permutation feature importance, which main idea is to compare for each feature i the accuracy of plain basic model with the accuracy of the model trained on the same feature set, but with randomly permuted values of i feature. SHAP[1] (SHapley Additive exPlanations) technique is expanding stated above permutation feature importance idea, based on computing Shapley values for permutation feature importance. It requires constructing a corresponding model for each subset of given features, which makes SHAP technique computationally hard. Similar, but more computational cost effective method is LIME[2] (Local Interpretable Model-Agnostic Explanations), which is basically similar to SHAP, but not all subsets of given features are taken into account. Other model-agnostic interpretation techniques include model class reliance (MCR)[3] technique, partial dependence plot[4], and technique based on partial dependence plots called individual conditional expectation (ICE) plots [5].

Model-specific solutions depend on exact model, so we are interested in the main techniques for random forest model. There are a lot of random forest interpretation techniques, which are not based on model approximation, such as LionForests[6], which provides multiple steps such as association rules, clustering and

random selection in order to interpret model results. Other interpretation technique for random forest is a method, based on feature contributions[7], which are basically the aggregated local increments for each tree. A local increment for a feature i represents the change of the probability of being in class C between the child node and its parent node provided that i is the splitting feature in the parent node.

There exist a bunch of interpretation techniques for random forests, which are based on approximating model with a different model, which corresponds the most to the goal of this paper. One of such techniques is inTrees[8], which extracts, measures, prunes and selects rules from a tree ensemble, and calculates frequent variable interactions, forming simplified tree ensemble learner. Other technique[9], proposed by Satoshi Hara and Kohei Hayashire, is based on constructing a new model with reduced number of decision regions, while minimizing such model error. In paper[10] by Ulf Johansson the interesting approach of approximating the random forest with single decision tree is used.

Also, some approaches use rule extraction algorithms, which is intended to be made in this paper. In Morteza Mashayekhi and Robin Gras work[11] the hill climb algorithm is used to search for rule sets and reduce the rules which has improved the comprehensiveness of the random forest. In other paper[12] by Sheng Liu and Shamitha Dissanayake, they have used a rule based regression algorithm that uses 1-norm regularized random forests to extracts a small number of rules from generated random forests and eliminates unimportant features.

Method, introduced in this paper belongs to random forest model-specific interpretation techniques, based on approximating initial model with another computed model.

3 METHODOLOGY

Given both training and test datasets on multiple features with categorical target variable, our goal is to construct a model on training set which improves the explainability and then to validate it on test set to compare with plain random forest model. The proposed explainability enhancing approach is divided into five consecutive steps (see Figure 1).

3.1 Step 1: Random Forest

For given training dataset with categorical target value, the random forest model is applied, leading to N different decision trees, which votes are aggregated to the final plain random forest prediction by simple majority voting. Each of these N decision trees are considered separately afterwards on steps 2 and 3 with purpose of independently extracting rules from each decision tree.

3.2 Step 2: Adding Target Variable

The subgroup discovery is applied to each decision tree in order to search for subgroups in data, which may be modelled well with considered decision tree (i.e. considered decision tree predictions on that subgroup are mostly accurate). Original target variable $Y_{pred}^{DT_k}$ of k-th decision tree is showing the predicted class, and not how good some prediction was, so it is unapplicable for the subgroup discovery which we need. That means that a new target variable should be introduced, which shows if the decision tree prediction is

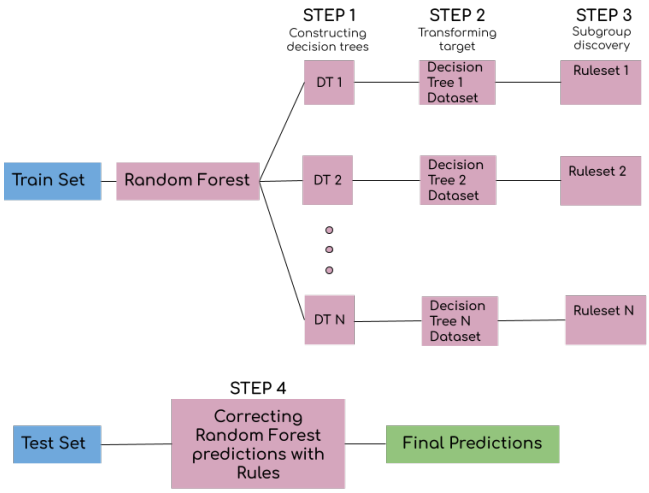

Figure 1: Step-by-step scheme of an algorithm

	Feature 1	...	Feature M	Y Train	DT (k) predictio n	DT (k) target
				A	B	A==B
Observation 1	...	...	...	Income >50K	Income <50K	False
Observation 2	...	...	...	Income >50K	Income >50K	True
Observation 3	...	...	...	Income <50K	Income <50K	True
...	...	...	...	...	...	...

Figure 2: Target variable transformation

the same as **(1)** random forest prediction Y_{pred}^{RF}, **(2)** ground truth, i.e. initial training dataset target variable Y. Each of these two options are possible and taken into consideration to validate, whichever option is more suitable for the initial goal (i.e performs best by accuracy).

On the Figure 2 one may find an example of k-th decision tree target variable transformation for the option **(2)**, resulting in the target variable value *True* if the initial training dataset target class is the same as the class predicted by decision tree, and *False* in case they are not. Resulting dataframe (noted as DF_k for k-th decision tree) consists of initial feature columns and transformed target column (marked with green on the example table) which are proceeded for the subgroup discovery step.

3.3 Step 3: subgroup discovery

Subgroup discovery is applied for each of N dataframes DF_k (which were obtained on the previous step), extracting the rules in the form of

$$R : Condition \rightarrow Target = True$$

Intuition for these rules are as follows: they state the regions of data limited with conditions, on which k-th decision tree predicts the same as **(1)** random forest prediction, **(2)** ground truth. The metrics used with these rules is the standard Chi-Squared metrics[13], allowing to range extracted rules by this metrics, showing the most trustable rules.

After the rules for each decision tree dataframe DF_k are extracted, they are aggregated in general table of rules, showing the most trustable rules among all N decision trees. This table consists of following information: rules, decision tree k corresponding to each rule and also the value of Chi-Squared metrics for each rule. Example of such a table may be found below.

Rule	According DT	Chi Squared metrics
$rule_1$	6	957.05
$rule_2$	4	935.92
$rule_3$	7	896.64
$rule_4$	6	891.41
...	...	...

3.4 Step 4: Correcting Predictions

Predictions of the random forest on the test dataset are corrected in a following way. The obtained rules are sequentially applied according to metrics (from highest to lowest), so if some observation fulfill the condition of the rule, then its prediction is changed from random forest-predicted class to the class predicted with decision tree corresponding to the applied rule. This change allows to count new prediction as explainable, because it was derived from single decision tree model, and not the random forest model as initial prediction for such observation. So, after applying the rule, some amount of observations are counted as explainable, and resulting accuracy for the new model is computed, allowing to explore the trade-off between amount of explainability of observations and the accuracy change.

4 RESULTS

The above described method was validated on Adult Data Set with 14 features, containing data on person and binary target variable showing this person's income. Training set consists of 32561 observations, and the test set consists of 16281 observations.

The code was implemented using Python 3 language, with following packages involved:

- *pysubgroup* for subgroup discovery;
- *sklearn* for random forest/decision tree fitting.

The method is not limited with exact random forest parameters, so for obtaining results we used *sklearn* library random forest with standard parameters, which may be found in Appendix A.

The results are divided in two parts according to the way of transforming target variable(see Subsection 3.2). The first method introduces introduces target variable according to initial random forest prediction, and the second uses the transformation according to the ground truth.

4.1 According-to-random-forest method

	accuracy	explainability	#rule
0	0.785763	0.000000	None
1	0.787114	0.377311	17
2	0.786930	0.392543	23
3	0.785824	0.399607	22
4	0.799459	0.399853	6
5	0.751674	0.883730	26
6	0.741539	0.884467	16
7	0.751858	0.915300	37
8	0.741109	0.915300	34

Figure 3: An example of algorithm run

Figure 3 shows results of applying top-50 obtained rules sequentially at random, until at least 0.9 of the dataset is explained (i.e. 0.9 of all predictions comes from single decision trees, not random forest model). After 0.9 explainability is increasing slow and may never reach 1.0, i.e. full explainability, so it is is found by us useless to explore values more than 0.9. First row corresponds to plain random forest model, and may be seen as a benchmark for accuracy. Third column shows the number of rules, sorted by metrics pool of rules (i.e. 0-th rule corresponds to the highest valued rule).

As may be seen from the table, one rule may cover more than 40% of the table, which makes whole process discrete and uneven. On the other hand, rules may intersect on some observations, which makes next rules explain only some small part of the dataset (because most of the observations, fullfilling the condition of such rule, already were explained).

This discreteness leads to the idea of averaging such runs. For each run we may linearly interpolate results, obtaining accuracy for each possible value of explainability we need. Then such runs are averaged, so the mean and standard deviation may be computed. After that we may compute confidence interval for mean.

Averaged results after 100 runs of the first method may be found on Figure 4. The accuracy of initial random forest, and the accuracy of average decision tree are taken as the baselines for accuracy.

4.2 According-to-ground-truth method

For these method same heuristics is kept, making validation process to be based on random sampling of the 20 strongest rules. Averaged results after 100 runs may be found in Figure 5.

4.3 Comparison of the methods

On Figure 6 one may found described above results on same plot, which makes possible the conclusion that the first model (which accords to predictions of random forest) is preferential.

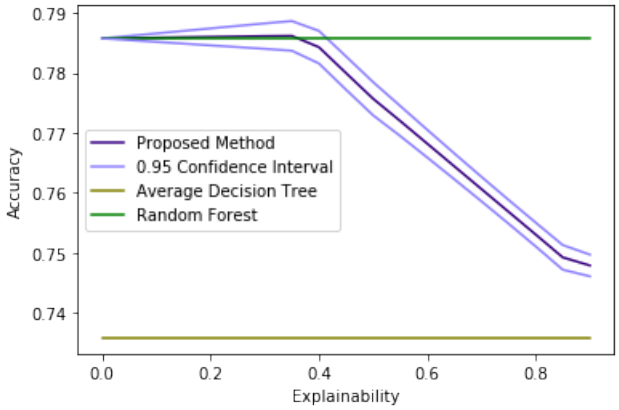

Figure 4: Method 1 averaged runs

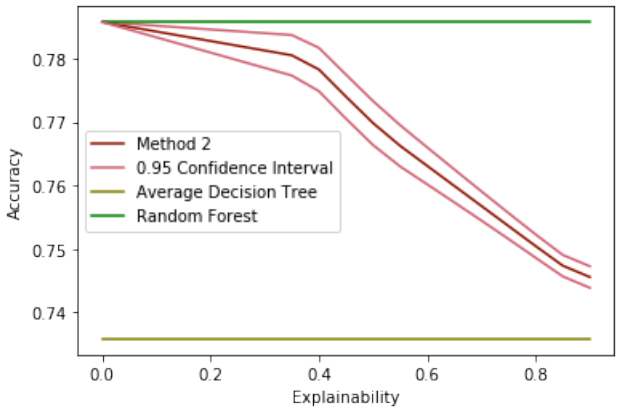

Figure 5: Method 2 averaged runs

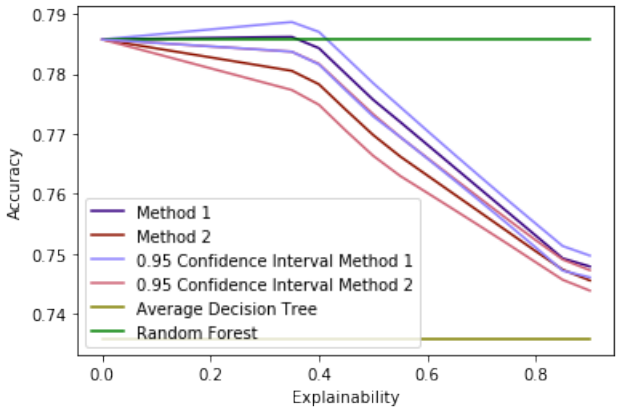

Figure 6: Method 1 and method 2 compared averaged runs

5 DISCUSSION

While accuracy of random forest model on the test set is 0.785 and the average accuracy of decision tree, of which the random forest consists, is 0.737, average accuracy for our model stays in this interval, making it more efficient than plain decision tree, but also allows to vary explainability-accuracy tradeoff, which makes this model useful.

From the above plots one may see that applying the first rule is usually more effective, then following. We state that based on the peak at the point where explainability 0.4, which may be seen on the graph. Usually first rule describes just the same (0.4) amount of the dataset.

The major drawback of this method is its discreteness, meaning that one cannot vary amount of explainability. For example, we cannot achieve less then 0.4 explainability with the rules mined on this dataset, because applying first rule usually already gives > 0.4 explainability.

That drawback makes a space for improvement of these method. The different metrics for the subgroup discovery might be explored, leading to less discrete variation of possible explainability amount.

A PARAMETERS OF USED RANDOM FOREST MODEL

RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini', max_depth=None, max_features='auto', max_leaf_nodes =None, min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=None, oob_score=False,
random_state=None, verbose=0, warm_start=False)

REFERENCES

[1] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. 2017. https://arxiv.org/pdf/1705.07874.pdf.
[2] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. *CoRR*, abs/1602.04938, 2016.
[3] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong but many are useful: Variable importance for black-box, proprietary, or misspecified prediction models, using model class reliance. https://arxiv.org/pdf/1801.01489.pdf.
[4] J. H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 1991. https://doi.org/10.1214/aos/1176347963.
[5] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. 2014. https://arxiv.org/pdf/1309.6392.pdf.
[6] Ioannis Mollas, Nick Bassiliades, Ioannis Vlahavas, and Grigorios Tsoumakas. Lionforests: Local interpretation of random forests. 2020.
[7] Anna Palczewska, Jan Palczewski, Richard Marchese Robinson, and Daniel Neagu. Interpreting random forest classification models using a feature contribution method. 2013.
[8] Houtao Deng. Interpreting tree ensembles with intrees. 2014.
[9] Satoshi Hara and Kohei Hayashi. Making tree ensembles interpretable. 2016.
[10] Ulf Johansson. One tree to explain them all. 2011.
[11] Morteza Mashayekhi and Robin Gras. Rule extraction from random forest: The rf+hc methods. 2016.
[12] Sheng Liu and Shamitha Dissanayake. Learning accurate and interpretable models based on regularized random forests regression. 2014.
[13] Tarek Abudawood and Peter Flach. Evaluation measures for multi-class subgroup discovery. 2016.