Skolkovo Institute of Science and Technology, Data Science Program

Report on Machine Learning course:

Hybrid of Regression Trees and Linear regression

Ivan Maksimov

Milena Gazdieva

Kristina Belikova

Mikhail Kuzin

Nikita Alexeichyk

SkolTech

2019

1. Introduction

In real world data is often linear within some subset, but non-linear in general. Indeed, we can recall "Simpson's paradox", according to which combining two sets with similar trends may lead this trend to turn into opposite or totally disappear.

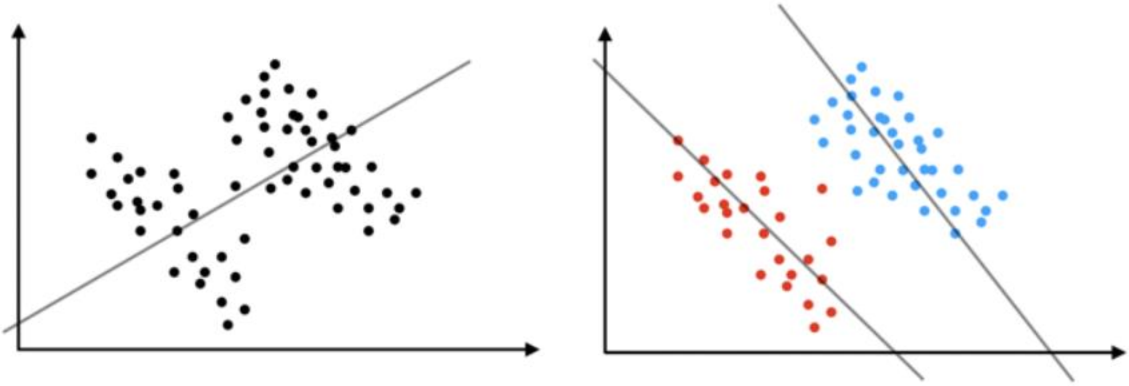


Figure 1.1 – Simpson’s paradox

Another example of this phenomenon is non-linearity within one feature. However, in many cases non-linear functions may be well approximated locally by a linear function. Example for this phenomenon is depicted at Figure 1.2

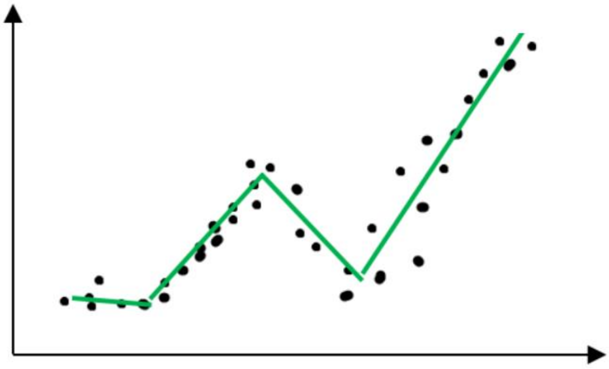


Figure 1.2 – Non-linear dependency, which is locally linear

Classic data science approach is to cluster data and fit a separate model per cluster or to use some methods that are non-linear as Random Forest or Gradient Boosting Trees. However, there are some disadvantages of these approaches.

For clustering and fitting a separate model per cluster true clusters are not known. Even their number is usually unknown. In addition, there are plenty of clustering algorithms and its hyperparameters, so it is hard to choose the “right” one. Moreover, data often has outliers, noisy or useless features and clustering is unsupervised algorithm – it is extremely hard to separate data in such subsample, so that error of the model(s) for entire dataset would be small.

As for RF and GBT, it is hard to learn linear dependency. Thus, if true dependency in data is linear (even within some subset) these methods may fail to learn it.

To overcome this issues it is possible to combine trees and linear models in order to learn linear dependency within some data subset. In the next sections corresponding algorithm, Hybrid of Regression Trees & Linear Regression, is discussed. Its theoretical motivation, algorithm itself, advantages and disadvantages are defined in section 2. Results of practical experiments, benchmark comparison and conclusions on its forecasting quality and efficiency are presented in section 3-5.

2. Related work

Basic notions and formulation of the Hybrid of Regression Trees & Linear Regression algorithm were introduced in [1] in 1992. While basic CART algorithm estimates value in the leaf of the tree as constant value, the idea of HRT algorithm is to use local linear regression in the leaves of the tree. The algorithm works in similar to CART way - it splits the example set, representing the node of the tree, into two subsets, from which it recursively builds subtrees. The HRT algorithm is depicted at Figure 2.

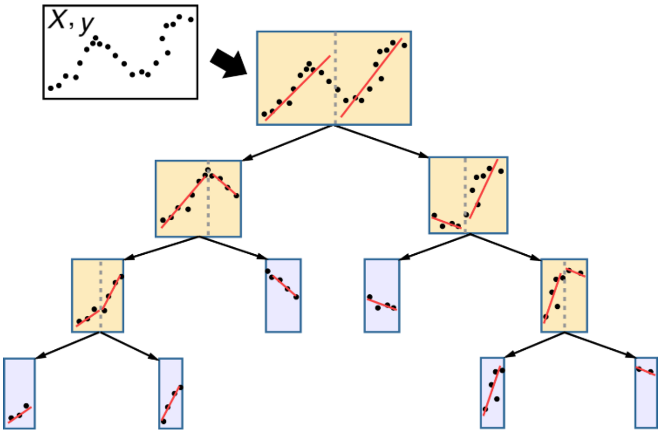


Figure 2. – HRT algorithm

As in the basic CART algorithm, the vital part of HRT is measure of goodness of the split, which is derived from the measure of the impurity of an example set. The best split of examples in a node is chosen in order to minimize expected impurity of the split:

$$I_{exp} = p_l I_l + p_r I_r, \tag{1}$$

where pl; pr - probabilities of transition into left/right nodes of the tree, Il; Ir - impurities of the corresponding example sets.

The impurity measure of an example set E is defined in the following way:

$$I(E) = \frac{1}{W(E)} \sum_{e_i \in E} (y_i - g(\overrightarrow{x_i}))^2, \qquad (2)$$

where W(E) - sum of weights of examples from E, function g represents the regression plane through the example set. One can note that in case of regression tree, impurity measure is mean squared error for “mean” prediction and in case of HRT – mean squared error of linear regression.

Advantages of HRT are:

- Explainability
- Non-linearity
- Requires small tree depth
- Can be run in parallel

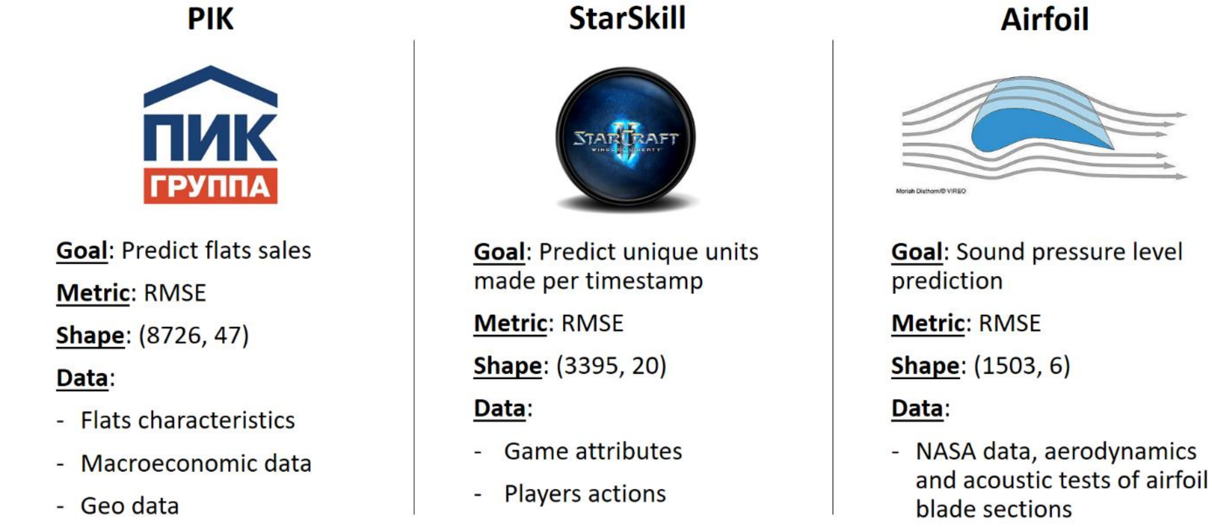
However, HRT also has some disadvantages:

- Is trained longer than linear regression as it fits many linear regression for finding best split
- Is trained longer that CART (for comparable parameters: max_depth, min_samples_leaf, min_impurity decrease) as HRT fits linear regression and CART just computes mean in the leaves
- Ensemble Hybrid is not explainable
- Poorly covers feature interaction
- Big variance for one HRT – better use ensembles of HRT

Further in this paper Bagging Ensemble of HRT is used to reduce the variance of predictions of one single Hybrid Regression Tree.

3. Dataset description

Bagging ensemble of HRT was applied to three datasets. Dataset names, target variable, forecasting quality metric, shape of data and a short data description are shown at Figure 3.



PIK

Goal: Predict flats sales

Metric: RMSE

Shape: (8726, 47)

Data:

- Flats characteristics
- Macroeconomic data
- Geo data

StarSkill

Goal: Predict unique units made per timestamp

Metric: RMSE

Shape: (3395, 20)

Data:

- Game attributes
- Players actions

Airfoil

Goal: Sound pressure level prediction

Metric: RMSE

Shape: (1503, 6)

Data:

- NASA data, aerodynamics and acoustic tests of airfoil blade sections

Figure 3. – Datasets description

4. Experiments

For each data set the following logic was applied:

1. Split data into train (70%) and test (30%)
2. Run GridSearchCV on train set, find best parameters according CV RMSE
3. Refit the model with the best parameters on the whole train set
4. Predict test targets
5. Measure error on test data set as RMSE

At Figure 4. The dependency of test RMSE on the share of data passed into the model with best hyperparameters (defined by GridSearch CV on train set) is shown. For this purpose only a fraction (30% – 70% with 10% step) was used. 70% corresponds to the full training sample. Note, that for all further experiments Bagging HRT ensemble was used,

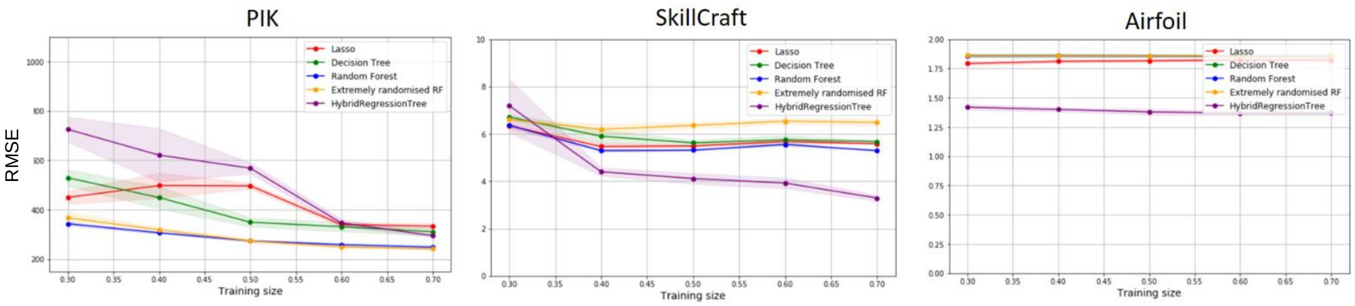


Figure 4.1 – Dependency of test RMSE on the share of data passed into the model with best hyperparameters

According to the experiment Bagging HRT ensemble outperforms other methods for 2/3 datasets. Particularly, for SkillCraft and Airfoil. For PIK dataset HRT RMSE improves dramatically as the training sample increases compared to other methods. So, probably, there are not enough training samples to outperform other methods. But still, for full training sample (70% of initial data set) its RMSE is lower, than Lasso and Decision Tree.

For HRT more significant dependency on the number of data it is trained on compared to other methods was seen for all of 3 data sets. Thus, HRT is sensitive to the number of data it is trained on. This result is in line with theoretical idea that linear regression requires pretty many training examples per feature. And HRT fits linear regression per leaf, so it requires pretty much training examples per leaf.

As it was defined during experiments, HRT is mostly dependent on two hyperparameters: **minimum number of examples per leaf** and **minimum impurity decrease**. The first hyperparameter controls number of samples on which final (in leaf) linear regression is trained on. Also it decreases a propensity to overfit to outliers. The second hyperparameter prevents further splitting when gain is small. Thus, it increases generalization ability (controls test set error). Maximum tree depth was no so important as two hyperparameters mentioned above are enough to stop splitting in a ‘right’ place. Indeed, controlling only tree depth, even for depth equal to one, in case of outliers one could get outliers in one leaf and all other data in the other. And in such a way overfitting to outliers may occur.

Figure 4.2 shows the dependency of training time on train size. Training time is log-scaled in order to make visualization more convenient.

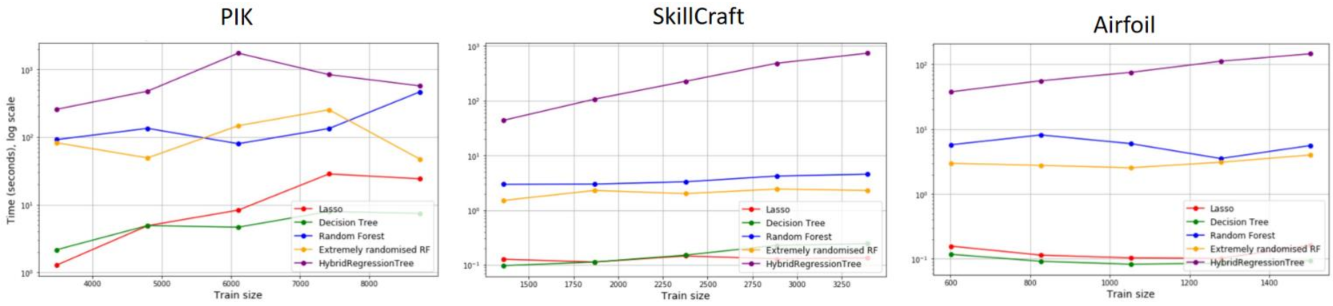


Figure 4.2 – Dependency of training time (log scale) on training size

As one could mention, training time for Bagging HRT is larger than all benchmarks for all datasets. Moreover, it is growing faster in sample size. Indeed, HRT is qubic in training sample, which is slower than all other mentioned benchmarks. As HRT requires pretty many examples to fit complex dataset (see RMSE for PIK) and its time complexity is big, it is mostly suitable for small datasets with strict big min samples per leaf stopping criterion and for medium size datasets.

5. Conclusion

Bagging HRT was benchmarked against Lasso, Decision Tree, Random Forest, and Extremely Randomized Forest on three datasets. The most important conclusions based on the analysis of results are:

1. Bagging HRT outperforms all mentioned benchmarks for some cases;
2. HRT is able to learn non-linear dependency in data;
3. Proposed algorithm is sensitive to the number of data it is trained on;
4. The most important hyperparameters are minimum number of examples per leaf and minimum impurity decrease;
5. Proposed algorithm has higher time complexity than benchmarks;
6. Bagging HRT is mostly appropriate for small-sized datasets with very strict stopping criteria and for middle-sized datasets

Further research includes comparison to gradient boosting regression trees, as it is one of the top performing algorithms in classic regression tasks. Also, as HRT is computationally expensive for practical use it is important to implement its parallelizable version.

6. References

1. Aram Karalic. Linear Regression in Regression Tree Leaves, Josef Stefan Institute, 1992.
2. Alexander K. SeewaldI, Johann PetrakI, Gerhard Widmer. Hybrid Decision Tree Learners with Alternative Leaf Classifiers: An Empirical Study, Austrian Research Institute for Artificial Intelligence, 2001
3. Tanujit Chakraborty, Ashis Kumar Chakraborty, Zubia Mansoor. A hybrid regression model for water quality prediction, Statistical Quality Control and Operations Research Unit, Indian Statistical Institute, 2003