Journal of Industrial Information Technology and Application *ISSN(Online): 2586-0852*
Vol.6 No.1

Decision Tree Development with Hierarchical Structure and its Optimization: A solution to overfitting

Yeheng Ma [1)], Sanghyuk Lee [1*)]
[1)] Department of Mechatronics and Robotics, School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou, China

Abstract. A decision tree design is carried out with the hierarchical structure. Specifically, K-fold and random forest algorithm is considered to overcome overfitting problem. Because of the existing problem –overfitting- is the most challenging together with performance when we consider decision tree building. Furthermore, big data and test data decision has always poor prediction together with complex models. One of the reasons could be due to the complex situation in real application cases compared to sample ones for training. The paper propose solutions to the overfitting issues of decision tree models. Simulation results are illustrated.

Keywords: Decision Tree, Overfitting, Random Forest, k-fold validation

1. Introduction

A decision tree is a divide-and-conquer approach to the classification which can be used to discover features and extract patterns [1]. Sometimes, it is challenge to show overfitting characteristics which makes poor prediction; for the new and exceptional data deteriorate its performance [2]. One of the reasons could be the complex situation in real application cases compared to sample ones for training. For example, financial market prediction or stock prices has nonlinear, complicated and stochastic property, so this would be poor prediction and affect in its precise and performance of designed models [3].

[*] Corresponding author: Sanghyuk.Lee@xjtlu.edu.cn
Received: Jan. 14, 2022; Revised: Feb. 5, 2021; Accepted: Mar. 31, 2022

Hence, we propose design decision tree that is complex and solve the financial datasets. Overfitting is a common problem in the topic of decision tree, and several methods have been used by researchers to overcome this. First, in terms of pruning, new research proposed an improved error-based pruning (IEBP), which uses k-fold cross validation and t-test to choose the optimal certainty factor to control the pruning process [5]. The proposed methods are composed of two steps; decision tree making with random forest (RF), heuristic approximation is also applied for optimization. There are two issues to be addressed: accuracy improvement, complexity advancement. To get optimized performance, we consider tree structure construction by RF as well as decision measure design for the training data. It also include curse of dimensional problem, and it was raised by Blanc *et al.* in 2020, which considers the correlations between the target function and small subsets of its attributes; it shows the comparison between Gini impurity and information gain that merely considers the correlations between the target function and an individual of its attributes [6]. Furthermore, RF also has a wide application to improve the accuracy of prediction, and the research are shown in the references [7-10]. Finally, there are innovative ways that leads to conduct early stopping. For instance, a novel cost function for stopping based on Kolmogorov complexity [2]. This study emphasize on hyper-parameter optimization for RF by random search and grid search.

Now a days, researches on machine learning (ML) to organize the decision tree models have become increasingly popular in financial market prediction [4]. However, complex models are often prone to overfitting and financial datasets tend to have low signal-to-noise ratio, therefore it is an imminent problem to automatically select effect features from datasets [4]. So, we focus on finding method to overcome the overfitting matter and proposes performance optimization through background research and model training experiments. Research outcomes are not limited to the financial markets, it can be applied to the decision and classification problems in more fields such as medical assessment, policy making and so on.

In this paper, two methods to solve overfitting are illustrated in the next section. Section 3 explains the industrial value of this research, and it clarifies the mythologies applied. Results for initial and optimized models are illustrated in Section 4, and discussion and analysis are also included. Finally, conclusions are included in Section 5.

2. Preliminaries

To overcome overfitting, two existing algorithm are introduced in this section: random forest and k-fold cross validation.

A. Random forest

Multiple decision trees are built by RF, which is constructed through bagging and it showed the effect solve overfitting performance [11]. Let f be the final prediction from the RF, B be the number of trees constructed from dataset, n be the index of decision tree constructed, f_n be the result from decision tree n, and x be the input sample that may not be in the training set. Final result of a prediction on a dataset can be expressed as eq. (1) [11]:

$$f = \frac{1}{B}\sum_{n=1}^{B}\left(f_n(x)\right) \tag{1}$$

In eq. (1), f_n represents the prediction of the n_{th} decision tree, which is a iterative process of assessment on selected features to a destination value. As Fig. 1 shows, RFs are chosen to hedge the risk of overfitting and inaccuracy from one decision tree, the choice of hyperparameters of which is conducted through random search illustrated in Fig. 1.

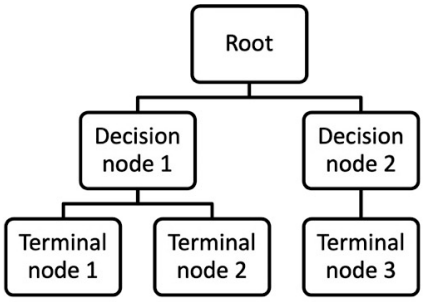

Fig. 1 Decision Tree

B. k-fold cross validation

The k-fold cross validation is applied on a dataset to deliver the errors of a model, by chunking the dataset into K chunks and using all of them as training and test examples [12].

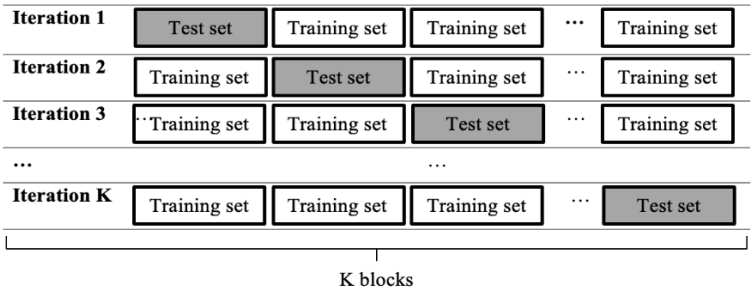

Fig. 2. K-fold cross validation

The cross-validation estimator is illustrated as the average of errors of all iterations [12]:

$$E(D) = \frac{1}{K}\sum_{n=1}^{K}\frac{1}{m}\sum_{z_i \in T_k} L(A(D_k), z_i) \quad (2)$$

where $D = \{z_1, z_2, z_3, \ldots z_k\}$ be the dataset applied in Fig. 2, $m = \frac{n}{K}$ be the size of each block, T_k be the k_{th} block, D_k be the training set excluding T_k. This study applies k-fold cross validation to deliver an averaged assessment result of predictors in attempts of random search. Random search explores the configuration space of parameters to find the best combination of them and find the model with highest accuracy [13]. Taking the time complexity under consideration, we combines random search with grid search to find the optimized parameters of the RF.

3. Decision Tree and RF

A. Decision Tree generation

We generate decision model with the help of decision tree and ensemble RF. First, we compares the accuracy of predictions on the dataset of a company's stock prices from Yahoo Finance. Table 1 shows the accuracies and Fig. 3 shows the comparison among prediction values from both models and actual values.

Table1. Accuracy Comparison

Number of training instances	Accuracy (%)		Mean Absolute Error (degrees)	
	Single decision tree	Random forest	Single decision tree	Random forest
450	99.72	99.69	0.9	0.97
900	99.64	99.58	1.22	1.45
8100	99.47	96.22	13.76	43.97

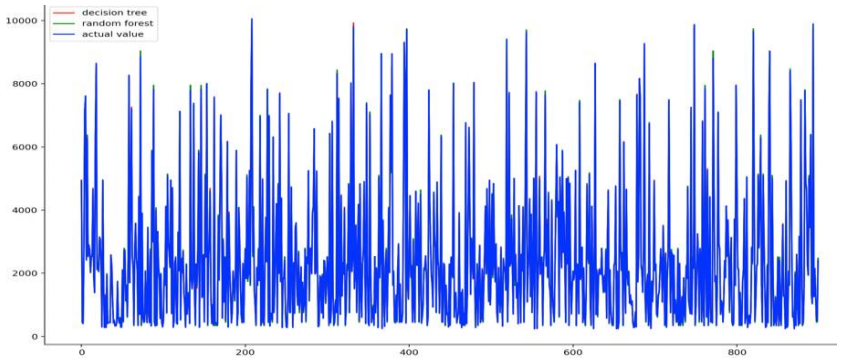

Fig. 3. Prediction values and actual values

Fig. 4. Poor predictions

By the comparison result in Table 1, two models perform well on training datasets, with accuracies over 98% and almost no difference between prediction and actual values. Besides the effort of models, the reason of such high accuracy might also be that when choosing training and testing samples the order of data is shuffled and thus the patterns of both datasets are similar, which barely affects the conclusion. However, when operating on real data which is out of training and testing samples, both models make poor predictions as Fig. 4 shows, which is referred to as overfitting. Dealing with large test datasets, the mean absolute error

of random forest increases sharply (from 1.45 to 43.97 degrees), it means the test results do not guarantee ideal predictions and that is where the model needs to be optimized.

B. Analysis on Performance

To optimize the model in proper dimensions, we focuses on the RF and analyze the factors behind its poor prediction performance.

- Data pattern

  As mentioned in before, the performance on test data is believed to be due to similar pattern between training and testing data. As shown in Fig. 5, the testing one after training are basically the same with many peaks, and that is one of the reasons why initial RF model behaves well on the test data. In comparison, the pattern of the real dataset is barely the case as shown in Fig. 6, with only one peak and be *centralized* at the peak, so the prediction behavior on these data is much worse than before. To further depict the difference of data pattern, Table 2 introduces the mean and standard deviation of each dataset.

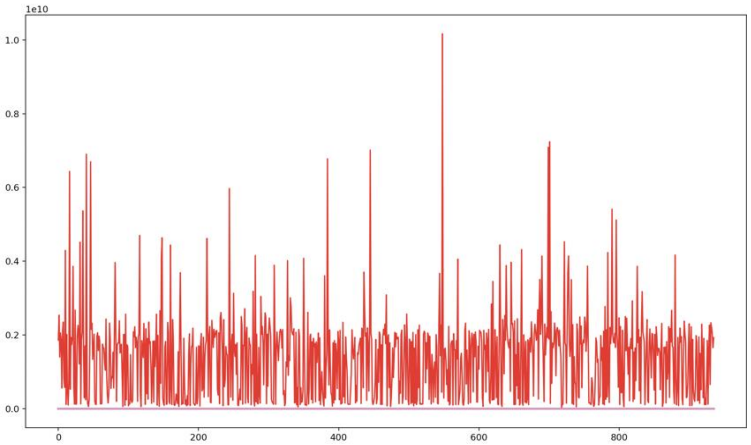

Fig. 5 Testing data

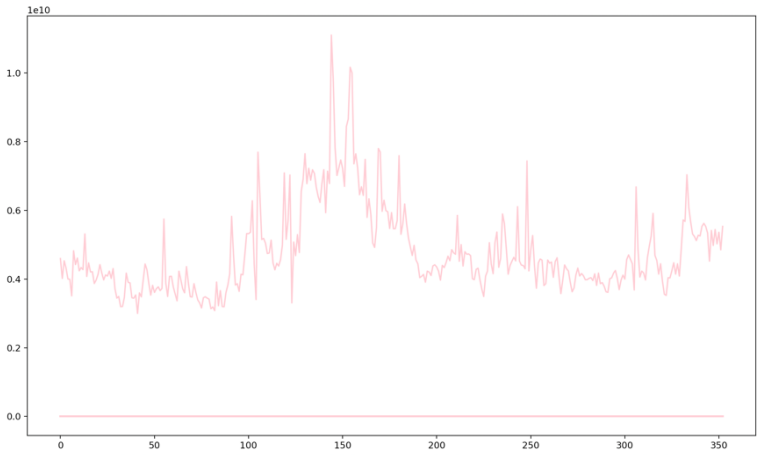

Fig. 6 Real data

Table 1. Mean and standard deviation of each dataset

Dataset	Mean	Standard Deviation
Training data	135922.32	16211997.89
Testing data	137044.19	16485129.28
Real data	510842.44	51096958.01

- Parameters

  According to the document of scikit-learn 1.0.2, the RF model is composed by around 100 decision trees with no specific maximum depth or minimum number of samples of a leaf [14]. These parameters are fixed and related to the complexity of the model, and either an over easy or over complex model would lead to poor predictions.

The change in data pattern and inflexible parameters are identified as two factors behind the sharp distinction between the prediction performance of the RF on different datasets. The solution of random search is applied to enable parameters to adapt to the different data patterns.

4. Optimized model and Simulation

To optimize the RF model, this section applies random search and grid search with random grid shown in Table 3 and grid based on random search shown in Table 4, with

reduced data sample (9000 instances to 5000 instances). In each random and grid search, the model applies k-fold cross validation by 3. The optimized parameters adopted by new RF model are concluded in Table 5. Fig. 7 shows the performance of the optimized model with real data.

Table 3. Random grid

Number of decision trees	10…200, number = 30
Max depth of decision trees	2, 3, 4 … 50
Min sample size to be spitted	[2, 5, 10]

Table 4: Grid for grid research

Number of decision trees	[81, 82]
Max depth of decision trees	[6, 7]
Min sample size to be spitted	[4, 5, 6]

Table 5. Optimized parameters

Number of decision trees	82
Max depth of decision trees	7
Min sample size to be spitted	5

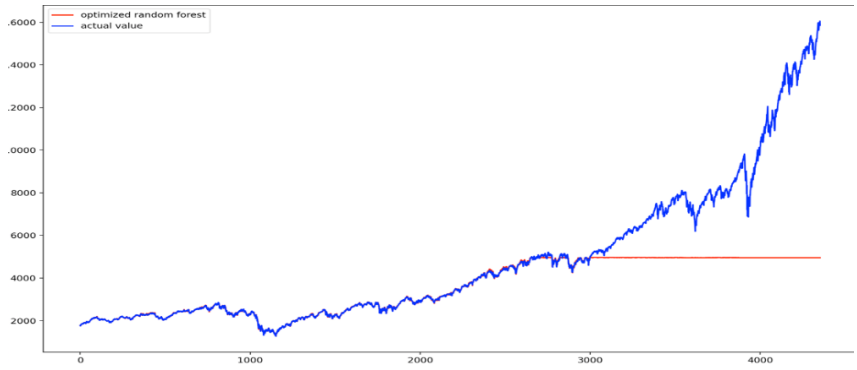

Fig. 7 Optimized prediction

After reducing the data size and applying random search, RF model shows better compared to the initial one, although there are poor predictions at the tail. The main distinction of the optimized model is that its complexity is adaptable to different patterns of datasets with random search and grid search. Additionally, small size of samples is enough. However, both models apply the same loss function and split criteria, as a result of which

there might still be some bottlenecks even though the parameters are comparably optimized, and that is the direction of further research.

5. Conclusions

This study has implemented the RF model on datasets of several levels and made a comparison with single decision trees, and improve the quality of the RF model to reduce the extent of its poor performance and the adaption to out-of-sample data. However, the time complexity of the algorithm is large, especially in the part of random search and grid search. The progress has moved into the core part of the study: overcoming the overfitting issues. Hence, it needs to adopt further approach to the overfitting problem with the data samples. Then, depending on the feasibility, the solution would be generalized to other and larger datasets and proposed from the algorithm perspective.

References

[1] J. Myles, R. N. Feudale, Y. Liu, N. A. Woody, and S. D. Brown, "An introduction to decision tree modeling," *Journal of Chemometrics*, vol. 18, no. 6, pp. 275-285, 2004, doi: 10.1002/cem.873.

[2] R. G. Leiva, A. F. Anta, V. Mancuso, and P. Casari, "A Novel Hyperparameter-free Approach to Decision Tree Construction that Avoids Overfitting by Design," ed, 2019.

[3] T.-Y. Hsu, "Machine learning applied to stock index performance enhancement," *Journal of Banking and Financial Technology: An Official Journal of the Institute for Development and Research in Banking Technology*, Original Paper vol. 5, no. 1, p. 21, 2021, doi: 10.1007/s42786-021-00025-6.

[4] C. Zhang, Y. Li, X. Chen, Y. Jin, P. Tang, and J. Li, "DoubleEnsemble: A New Ensemble Method Based on Sample Reweighting and Feature Selection for Financial Data Analysis," 2020, Conference. [Online]. Available: http://login.ez.xjtlu.edu.cn/login?url=https://search.ebscohost.com/login.aspx?direct=true &db=edseee&AN=edseee.9338413&site=eds-live&scope=site

[5] Y. Peng, Y. T. Lu, and Z. G. Chen, "An Improved Error-Based Pruning Algorithm of Decision Trees on Large Data Sets," in 2021 IEEE 6th International Conference on Big Data Analytics (ICBDA), 5-8 March 2021 2021, pp. 33-37, doi: 10.1109/ICBDA51983.2021.9403001.

[6] G. Blanc, N. Gupta, J. Lange, and L.-Y. Tan, "Universal guarantees for decision tree induction via a higher-order splitting criterion," ed, 2020.

[7] A. L. Boulesteix, S. Janitza, J. Kruppa, and I. R. König, "Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 2, no. 6, pp. 493-507, 2012.
[8] Y. Qi, "Random forest for bioinformatics," in Ensemble machine learning: Springer, 2012, pp. 307-323.
[9] G. Biau and E. Scornet, "A random forest guided tour," Test, vol. 25, no. 2, pp. 197-227, 2016.
[10] M. Belgiu and L. Drăguţ, "Random forest in remote sensing: A review of applications and future directions," ISPRS journal of photogrammetry and remote sensing, vol. 114, pp. 24-31, 2016.
[11] G. Rebala, A. Ravi, and S. Churiwala, "Random Forests," in An Introduction to Machine Learning. Cham: Springer International Publishing, 2019, pp. 77-94.
[12] Y. Bengio and Y. Grandvalet, "No unbiased estimator of the variance of k-fold cross-validation," Journal of machine learning research, vol. 5, no. Sep, pp. 1089-1105, 2004.
[13] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," Journal of machine learning research, vol. 13, no. 2, 2012.
[14] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.