Project 3: Decision Tree: ID3 and CART

Dongjun Cho DCHO13@JH.EDU
Programming Project 2
605.649
13 March 2022

Abstract

This project is intended to implement the K-Nearest Neighbor Algorithm for classification and regression. This algorithm was applied to discrete and continuous-valued data using six data sets obtained from the UCI Machine Learning Repository.

1. Introduction

The decision tree is a supervised learning machine learning model that allows us to find patterns from the attributes of each data. It connects the observed value and the target value for a certain dataset. By analyzing the data, the pattern existing between them is expressed as a combination of predictable rules. It's a concept similar to the Twenty Questions game where we narrow down the target by asking questions. Decision trees can be used for classification and regression.

This project is sought to implement Iterative Dichotomiser 3 (ID3) for classification tasks, and Classification and Regression Tree (CART) for regression tasks. These algorithms work greedily by dividing data into groups and creating rules for each division to predict the outcome class. When these algorithms construct the decision tree, there are four components: a root node, branches, leaf nodes. The root node is the starting nodes points in the tree. Each node represents the attributes of the dataset. Branches are decision rules that connect to leaf nodes. Leaf node represents the class label/target value.

The main problem of the decision tree is overfitting. It may overfit because splitting favors attributes with many values. If there are many branches in the tree, the impurity can be much less Alpaydın (2020). Also, it could end up generalizing on unseen data and learning noise. It can be avoided by using the pruning process. There are common techniques to process the pruning: reduced error pruning, and early stopping. Both pruning techniques can be applied to both classification and regression. For this paper, reduced error pruning is applied for ID3, and early stopping is applied to CART. Since the purpose of pruning is to avoid overfitting the data, pruned decision tree will outperform an unpruned tree.

Hypothesis 1: We hypothesize that pruned decision trees will outperform unpruned decision trees. To test this hypothesis, we compared the accuracy score for classification and mean squared error for regression tasks on several datasets from the UCI repository.

In Section 2, we describe how each algorithms works, in section 3, we describe the experimental approach to the algorithm, in section 4, we present the result of our experiment. In section section 5, it discussed about the relationship between performance and distribution of the data. We discussed the behavior of the algorithm in section 6, and then we conclude the discussion with the result and possible future works in section 7.

2. Description of Algorithms

2.1 ID3

ID3 is a non-parametric supervised learning algorithm that uses the entropy reduction in Shannon's information theory. It recursively builds the decision tree by selecting the best attribute based on the information gain ratio. It computes the entropy (impurity) that measures how much uncertainty is in the data.

$$I(c_1,\ ...,\ c_k) = Entropy = -\sum_{l=1}^{k} \frac{c_l}{c1+...+c_k} \log \frac{c_l}{c1+...+c_k}$$

To compute expected entropy, it multiplies the ratio of the total points in the partition by the entropy of the partition. It computes the expected (average information) entropy using the formula below,

$$E(f_i) = \sum_{j=1}^{m_i} \frac{c^j_{\pi,1}+...+c^j_{\pi,k}}{c_{\pi,1}+...+c_{\pi,k}} I(c^j_{\pi,1}, ...,\ c^j_{\pi,k})$$

For categorical attributes, it determines the best optimal splits by computing the gain ratio for each attribute. It computes the gain ratio by subtracting entropy to expected entropy.

$$gain_\pi\left(f_i\right) = I\left(p_\pi, n_\pi\right) - E_\pi\left(f_i\right)$$

For numerical attributes, it determines the best optimal splits by determining all possible binary splits at midpoints between adjacent data points. To determine possible binary splits, it first sorts the dataset by the class label. Then, it chooses all possible binary splits points where the class label change occurs. For each binary split point, it computes information gain by subtracting entropy to expected entropy. (For this project, I was able to figure out how to get best attribute using numerical, but I wasn't able to implement the decision tree with numerical attribute.)

For categorical and numerical attributes, the attribute that has the highest gain ratio is set to the root node in the decision tree. It keeps growing trees until there is no feature attribute left. If every instance has the same class or no more instances left, it chooses the most common class.

2.2 CART

Unlike ID3, CART is represented as a binary tree, and it uses MSE instead of entropy.

$$MSE(x) \ = \ \frac{1}{n}\sum_{i=1}^{n}\left(x-\bar{x}\right)^2$$

To determine possible binary splits, it distributes the dataset using the equal frequency method. Then, it chooses possible binary splits points where the frequency of class labels changes. It splits the points that minimize the attribute's mean squared error. For constructing a binary tree, the data that is less than the selected binary split is set to the left subtree. If the data is greater than equal to selected binary splits, it is set to the right subtree. It keeps partitions the data based on the selected attribute until there is no more data left to split.

3. Experimental Approach

3.1 Pruning

For this project, it used reduced error pruning to avoid over-fitting data for classification tasks (ID3). Since the pruning starts after the tree has grown to completion, it used post-pruning process. For each node, it tagged for pruning, then compare the performance of

untagged tree and tagged tree. If the tagged tree performs better than untagged tree, then tagged tree becomes new tree. But, it doesn't remove a node that decreases the accuracy. It continues to prune the node until the performance of the resulting pruned tree has no further improvement occurs relative to the original tree.

It used early stopping to avoid over-fitting for regression tasks (CART). Given the threshold, it keeps growing the tree until there are no further splits made or the gain ratio (MSE) associated with the node drops below than than the threshold. If feature MSE is less than the threshold value, it will not proceed further and returns a mean of the remaining points.

3.2 Tuning

For the tuning process, it only applies to regression tasks, not classification tasks. For the CART algorithm (with early stopping), there are several parameter factors that need to consider. There is three-parameter that needs to consider, Equal Frequency size, Minimum Leaf to grow, and Threshold. The size of equal frequency determines how many splits are best when we are splitting the attributes. In CART, there are two rules to stop growing trees: minimum leaf number, and threshold. The minimum leaf determines the maximum number of the leaf that can grow in the decision tree. Threshold determines the error threshold.

For this implementation, the size of equal frequency ranges from 2 to 10. The number of leaf limits ranges from 2 to 10. For threshold, it sets as $\{0.2,\ 0.4,\ 0.6,\ 0.8,\}$. To tune these parameters, the total dataset is split into training (80%) and tuning (20%). Using 20% of the dataset, it processes the tuning to find the best parameter combination. Once it finds the best parameter, it is applied to the training set. With the best parameter values, the training set is applied to 5-fold cross-validation.

4. Results of Experiments

For classification, the table showed the accuracy score for the unpruned and pruned decision tree. For regression, it shows the best parameter and the mean squared error for unpruned and pruned decision trees.

4.1 Breast Cancer Dataset

The Breast Cancer set represents a classification problem used to distinguish whether it is benign or malignant based on 10 feature attributes. Table 1 is demonstrated the accuracy score for unpruned and pruned decision tree.

Table 1: Accuracy Score of Unpruned & Pruned

	1	2	3	4	5	Avg
Unpruned	80.36%	93.69%	77.48%	93.69%	93.69%	87.78%
Pruned	73.21%	90.09%	69.37%	87.39%	89.19%	81.85%

4.2 Car Evaluation

The Car Evaluation set represents a classification problem used to distinguish car acceptability based on 6 feature attributes. Table 2 is demonstrated the accuracy score for unpruned and pruned decision tree.

Table 2: Accuracy Score of Unpruned & Pruned

	1	2	3	4	5	Avg
Unpruned	83.03%	71.74%	79.35%	70.65%	69.2%	74.79%
Pruned	85.02%	79.35%	66.67%	65.58%	60.14%	71.38%

4.3 Congressional Vote

The Congressional Vote set represents a classification problem used to distinguish house party (republican/democrat) based on 16 features of attributes of legislation votes. Table 3 is demonstrated the accuracy score for unpruned and pruned decision tree.

Table 3: Accuracy Score of Unpruned & Pruned

	1	2	3	4	5	Avg
Unpruned	97.14%	92.86%	94.29%	92.86%	94.12%	94.25%
Pruned	97.14%	92.86%	94.29%	95.71%	92.65%	94.53%

4.4 Abalone

The Abalone set represents a regression problem used to distinguish the associate age of abalone based on 8 features attributes. Table 4 demonstrated the mean squared error for unpruned and pruned decision tree.

Table 4: MSE of Unpruned & Pruned

	Freq	Leaf limit	θ	1	2	3	4	5	Avg
Unpruned	2	-	-	10.86	6.04	4.84	4.45	35.08	12.25
Pruned	2	6	0.2	11.78	6.15	5.09	4.54	35.09	12.53

4.5 Computer Hardware

The Computer hardware set represents a regression problem used to distinguish the CPU performance based on 10 features attributes. Table 5 demonstrated the mean squared error for unpruned and pruned decision tree.

Table 5: MSE of Unpruned & Pruned

	Freq	Leaf limit	θ	1	2	3	4	5	Avg
Unpruned	3	-	-	0.8	0.16	0.23	0.48	5.35	1.4
Pruned	4	2	0.2	1.16	0.21	0.27	0.75	5.09	1.5

4.6 Forest Fires

The Forest Fires set represents a regression problem used to predict the burned area of forest fires by using 12 attributes features of meteorological data. Table 6 demonstrated the mean squared error for unpruned and pruned decision tree.

Table 6: MSE of Unpruned & Pruned

	Freq	Leaf limit	θ	1	2	3	4	5	Avg
Unpruned	9	-	-	3.68	4.08	2.83	1.32	8.0	3.98
Pruned	2	7	0.8	2.26	2.72	1.15	0.95	8.37	3.09

5. Distribution of the data

I found interesting points for the score for each fold. For classification, the accuracy score for each fold is evenly distributed. But, the MSE for regression for each fold is showing uneven especially in the fifth fold compared to other folds. Based on the previous assignment (project 2), the regression's cross-validation samples uniformly across all the response values. So, it sorts the data by the predictor, then takes the fifth point for a given fold. Based on names fileDua and Graff (2017) for regression tasks, the dataset is distributed unevenly. The table below shows how the unique value of predictor is distributed in each fold for Abalone and Computer Dataset.

Table 7: Unique attribute for Abalone & Computer Dataset for Each Fold

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Abalone	$1 \sim 7$	7, 8, 9	9, 10	10, 11, 12	$12 \sim 29$
Computer	0, 1	1	1	1, 2	2 $\sim$7

Based on the names fileDua and Graff (2017) for abalone and computer dataset, it shows that data is heavily distributed in the class $6 \sim 12$ in the abalone dataset. Also, the data is heavily distributed in the second class, 1 for the computer dataset. Based on the result of the experiment, it shows the MSE reduces if there is a small number of the unique attribute to predict. So, we know that the fifth fold score is uneven compared to other fold scores because there is a large number of the attribute to predict.

6. Behavior of Algorithms

The performance of the unpruned decision tree and pruned decision tee was not expected. Hypothesis 1 stated that the pruned tree would outperform the unpruned tree. Throughout the 6 datasets (3 classifications, 3 regression), 2 datasets match with the hypothesis. For overall performance, the congressional vote performs the best compared to other dataset. For classification, the Unpruned decision tree outperformed pruned tree. But, the Congressional vote dataset shows pruned tree is slightly better than the unpruned tree.

For regression, it showed the similar result to classification tasks. Unpruned decision tree outperformed pruned tree as well. But, the pruned tree performed better than the unpruned tree on the forest dataset. For overall performance, the computer dataset performs the best compared to other dataset. It is very interesting that it showed that the pruned tree outperformed unpruned tree for one dataset from classification, and one dataset from regression.

The purpose of the pruning process is to avoid over-fitting the data. However, Since the pruning process shortens the decision tree by keeping the node that decreases accuracy, it

could lead to under-fitting and lower accuracy for the test dataset. It is very interesting that pruning does not necessarily guarantee better performance.

7. Conclusion

This paper implemented the two types of the decision tree: Iterative Dichotomiser 3 (ID3) for classification tasks, and Classification and Regression Tree (CART) for regression tasks. Throughout this paper, I was able to learn how the decision tree tries to prune the node to avoid over-fitting. I learned that pruned trees don't necessarily guarantee better performance. I also learned that distribution of data does effect the performance score for the model.I think the feature importance selected by each algorithm can help to determine the most important feature variable for the problem.

References

Ethem Alpaydın. *Introduction to Machine Learning, fourth edition.* The MIT Press, 2020.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.