Journal of Machine Learning Research 1 (2020) 1-7 Submitted 10/11

Classification and Regression Decision Trees

Jieyi Chen

October 11, 2020

1. Introduction

This project explores the application of classification and regression trees on six datasets from the UCI Machine Learning Repository [1]. Specifically, we apply the Iterative Dichotomiser 3 (ID3) algorithm on the classification dataset and Classification and Regression Tree (CART) algorithm on the regression dataset. These nonparametric algorithms have no assumption on the parameters of the population. The main concept of the algorithm is to create a decision tree recursively until the stopping criteria is met. A decision tree has three components: the node, the branch and the leaf node. Each node represents an attribute, the link indicates the decision rule that is applied, and the leaf node is the class label or predicted outcome. Moreover, we explore the concept of pruning, a common strategy to optimize a decision tree by removing nodes that do not provide additional information. The common pruning processes are pre-pruning (early stopping) and post-pruning (reduced-error pruning). Pre-pruning is a technique that stops growing the tree when it is considered efficient. Post-pruning is a process that simplifies the tree after it has fully grown. Both pruning methods can be applied to classification and regression datasets. For this project, we implement reduced error pruning with ID3 and pre-pruning with CART. We hypothesize that the pruned trees outperform the unpruned trees for both algorithm because the goal of the pruning is to make the tree more generalized so that it does not overfit and have higher accuracy in the unseen data. To test this hypothesis, we compare the performances of ID3 and CART algorithms with and without pruning on several datasets from the UCI repository.

The rest of the paper is organized as follows. Section 2 describes the algorithms that are implemented. Section 3 discusses the experimental approach. Section 4 presents the results of the experiments. Section 5 discusses the results. Section 6 concludes.

2. Algorithm Implementation:

2.1 Iterative Dichotomiser (ID3) Algorithm:

ID3 is a supervised learning algorithm developed by John Ross Quinlan [2] and uses the notion of entropy reduction in Shannon's Information Theory. It recursively builds the tree with the attribute found using information gain until the stopping criteria is met.

Algorithm:

1. Calculate the entropy of the current partition of the dataset
2. For each available attribute in the dataset:
    a. For each available categorical feature in the dataset:
        i. Calculate the gain ratio = (current entropy – expected entropy)/intrinsic value
    b. For each numerical feature in the dataset
        i. Sort the data on that attribute, carrying the class label with it
        ii. Find all the possible binary split points, which are the midpoints between the points where the class change occurs

iii. At each binary split point, calculate its corresponding information gain = current entropy – expected entropy

3. Find the categorical attribute or split on the numerical attribute that maximizes the gain ratio or information gain
4. Partition the dataset based on the selected attribute and repeat steps 1-3 until it meets the stopping criteria

Concept of Entropy and Information Gain:

For datasets with 2 classes:

1. Measure I(p, n): the amount of uncertainty (entropy) in the dataset

$$\text{I(p, n) = entropy} = -\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$

2. Calculate the average information entropy

   Subset $D_\pi \subseteq D$

$$E_\pi(f_i) = \sum_{j=1}^{m_i} \frac{p_\pi^j + n_\pi^j}{p_\pi + n_\pi} I(p_\pi^j + n_\pi^j)$$

$$m_i = |d(f_i)| \qquad p_{\pi,}\ n_\pi \sim D_\pi$$

   where m_i is the size of the domain for feature i

3. Calculate the information gain: The difference in the prior entropy of the data before any splitting and the remaining entropy from dividing the data using feature i

$$\text{Information Gain } \pi(f_i) = I(p_\pi, n_\pi) - E_\pi(f_i)$$

Generalize these results for more than 2 classes:

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

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

$$c_{\pi,i}^j = \#\ examples\ in\ j^{th}\ partition\ from\ feature\ f_i\ with\ class\ c_k$$

In order to fight against overfitting, Quinlan proposes the gain ratio criterion, a measure that penalizes the attributes that would create a larger number of partitions.

$$\text{N}(f_i) = -\sum_{j=1}^{n_i} \frac{(p_i + n_i)}{(p+n)} \log\left(\frac{p_i + n_i}{p+n}\right)$$

$$\text{Gain Ratio} = \frac{\text{Gain}(f_i)}{IV(f_i)}$$

Stopping Criteria:

1) No more instances left in the subset; the leaf node is labeled with the class with the plurality vote in the parent node
2) Every instance in the subset is in the same class, the leaf node is labeled as the class of the instances
3) There are no more attributes left for splitting, the leaf node is labeled as the class with the plurality vote

2.2 Classification and Regression Tree (CART) Algorithm:

CART is a supervised learning algorithm introduced by Leo Breiman [3]. It is a recursively binary splitting process that partition the dataset. It lines up all the values, identifies the split points and select the point based on a cost function, which is mean square error in this case.

Algorithm:

1. Calculate the Mean Square Error (MSE) of the current partition of the dataset
2. For each available attribute in the dataset:
    a. For each available categorical feature in the dataset:
        i. The split points are the unique values of the feature
    b. For each available numerical feature in the dataset:
        i. Get the unique values of the feature and sort it from the smallest to the largest
        ii. Break these unique values into bins and consider the boundary points between these bins as potential binary split points
    c. At each binary split point, calculate its corresponding mean square error
3. Find the split point on the categorical and numerical attribute that minimizes the mean square error
    a. For the categorical feature, the left subtree contains data that contains the selected binary split point, the right subtree contains instances that do not have the binary split point
    b. For the numerical feature, the left subtree contains data that is less than or equal to the selected binary split point, the right subtree contains instances that is greater than the binary split point
4. Partition the dataset based on the selected attribute and repeat steps 1-3 until it meets the stopping criteria

Cost Function:

$$Mean\ Sqaure\ Error\ (MSE) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Stopping Criteria:

1) No more instances left in the subset; the leaf node is labeled with the average of the target value in the current partition
2) Every instance in the subset has the same target value, the leaf node is labeled as the target value
3) There are no more attributes left for splitting, the leaf node is labeled as the average of the target value in the current partition

Performance Evaluation Metric:

Classification:

$$\text{classification error } rate = \frac{\#\ of\ instances\ that\ are\ misclassified}{\#\ of\ instances\ in\ the\ test\ set}$$

Regression:

$$Mean\ Sqaure\ Error\ (MSE) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

3. Experimental Approach:

Datasets:

The datasets are from UCI Machine learning Repository [1]. Table 1 summarizes the properties of the six datasets.

Dataset Property Summary Table						
	Breast Cancer Dataset	**Car Evaluation Dataset**	**Image Segmentation Dataset**	**Abalone Dataset**	**Computer Hardware Dataset**	**Forest Fires Dataset**
# Attributes	9	6	18	8	7	12
# Examples	699	1,728	210	4,177	209	517
Type of dataset	Classification	Classification	Classification	Regression	Regression	Regression

Table 1 - Dataset Property Summary Table

Data Preprocessing Steps:

The only data preprocessing for this project is replacing the missing values with the average value of the attribute in the class for the breast cancer dataset. No preprocessing is needed for other datasets, which is one of the advantages of decision tree.

Pruning Process:

Pruning is a regularization technique that is used to optimize the decision tree and solve the overfitting problem. A decision tree T overfits the training set if the performance of tree T on the training set is better than its performance of tree T'. But the error of tree T on the test set is worse than tree T'. In other words, although the algorithm performs very well in the training set, it does not generalize well on the unseen data.

Post-Pruning/Reduced Error Pruning:

After we build a complete tree with the training set, we start the post pruning process with the help of the validation set. Each vertex of the tree is now a candidate for pruning. Using the bottom up approach, the selected nodes and its associated subtrees will be removed from the tree and assigned a leaf node with the majority class of the removed subtree. We continue the pruning process until the classification accuracy from the removal of the subtree has no further improvement on the validation set.

Pre-Pruning/Early Stopping:

This method sets a threshold and uses it to evaluate the mean square error to determine whether the error is in the acceptable criterion range. We stop growing the tree when the mean square error drops below the threshold.

Parameter Tuning Process:

Number of intervals for numerical attribute:
In the regression tree building process, we need to break the numerical feature into bins and consider the boundary points between the bins as potential binary split point. We need to the tune the number of intervals that we are considering. For this project, we decide that 4, 6 and 10 number of intervals are appropriate and select the one with the least mean square error. The selected number of bins are listed in Table 4.

Threshold for mean square error:
In the Pre-pruning process, we need to use a threshold to evaluate whether we should stop growing the trees. We need to tune the threshold parameter on the tuning set in order to determine the right threshold for stopping. The threshold is set to be 30%, 50% and 70% of the original mean square error (Total MSE) of the target value before any splitting. In other words, if the MSE is less than X% of the Total MSE, we stop growing the tree.

4. Result

Classification			
ID3 Algorithm Performance (Classification Error)			
Dataset	**Breast Cancer**	**Car**	**Segmentation**
Unpruned – fold 1	0.057	0.101	0.071
Unpruned – fold 2	0.043	0.090	0.143
Unpruned – fold 3	0.057	0.084	0.119
Unpruned – fold 4	0.050	0.058	0.143
Unpruned – fold 5	0.067	0.090	0.119
Unpruned – Average	**0.054**	**0.084**	**0.119**
Pruned – fold 1	0.092	0.168	0.095
Pruned – fold 2	0.035	0.147	0.143
Pruned – fold 3	0.050	0.139	0.167
Pruned – fold 4	0.071	0.118	0.214
Pruned – fold 5	0.126	0.128	0.143
Pruned – Average	**0.075**	**0.14**	**0.152**

Table 2 - ID3 Algorithm Performance - Classification Error

Regression			
Cart Algorithm Performance (Mean Square Error)			
Dataset	**Abalone**	**Machine**	**Forestfires**
Unpruned – fold 1	8.24	2,230	6,394
Unpruned – fold 2	6.90	2,997	7,742
Unpruned – fold 3	6.41	1,727	9,518
Unpruned – fold 4	6.79	3,396	3,542
Unpruned – fold 5	5.90	2,628	23,765
Unpruned – Average	**6.85**	**2,596**	**10,192**
Pruned – fold 1	11.74	7,029	628
Pruned – fold 2	6.90	27,914	6,303
Pruned – fold 3	6.41	35,142	11,795
Pruned – fold 4	6.79	38,447	1,117
Pruned – fold 5	5.90	19,245	534
Pruned – Average	**7.55**	**25,555**	**4,075**

Table 3 - CART Algorithm Performance - Mean Square Error

Number of intervals for numerical attribute (Mean Square Error)			
Dataset	**Abalone**	**Machine**	**Forestfires**
# of intervals	n = 10	n = 4	n = 6

Table 4 - Number of intervals for numerical attribute - Mean Square Error

Threshold for mean square error			
Dataset	**Abalone**	**Machine**	**Forestfires**
Threshold Selected	Threshold = 30%	Threshold = 50%	Threshold = 30%

Table 5 - Threshold for mean square error - Mean Square Error

5. Discussion:

Table 2 shows the performance of the ID3 algorithm using classification error. The breast cancer dataset has the highest accuracy, followed by the car and segmentation dataset. Table 3 shows the performance of the CART algorithm using mean square error. Table 4 and 5 show the optimal number of intervals/bins for the numerical attribute and threshold for the mean square error during early stopping. We hypothesize that the prune tree performs better on the test set. However, our experiment shows that it is not always the case. We can only guarantee the pruned trees perform better on the validation set, but not always on the test set. Our results indicate that the unpruned trees on average have higher accuracy on the test set for classification. For regression, the unpruned trees perform better than the pruned trees on the Abalone and Machine dataset, but the pruned trees outperform on the Forestfires dataset. We believe this happens because the depth of the tree can significantly influence its performance. Although the prune trees are more generalized, it is usually much shorter than the unpruned trees, which can lead to lower accuracy and underfitting. The tree's performance also varies on the dataset and the threshold that we set. Therefore, a better selection of thresholds can be done in future work.

6. Conclusions:

This project helps us gain a better understanding of decision trees and their performance on both classification and regression datasets. In addition, we learn two tree pruning methods that aim to make the tree more generalized and reduce overfitting. This paper can help analysts to decide which algorithm to implement when they encounter classification or regression problems. We have learned that the pruned trees do not necessarily guarantee better performance on the test sets. It depends on the nature of the dataset, the distribution of the target values and the thresholds that we select. Some future work on how to optimize the pruned trees can be done. It would also be interesting to apply the CART algorithm on the classification datasets and see whether it outperforms the ID3 algorithm.

7. Reference:

[1] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[2] Quinlan, J. R. 1986. Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. 1986), 81–106

[3] Breiman, Leo; Friedman, J. H.; Olshen, R. A.; Stone, C. J. (1984). Classification and regression trees. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software. ISBN 978-0-412-04841-8.