COMPX521: Assignment 1

Connor Welham

April 21, 2022

1 Introduction

M5, also known by its improved variant M5', is a decision tree learner for regression task. It is used to predict values of numerical response y. It is a binary decision tree which has linear regression models at its leaf nodes which predicts numerical attributes. However, M5 has an issue that it does not consider the linear regression models. This is an issue as the selection splits are selected based on variance reduction, but assume that constant predictors in the two branches are being created by a split. This assumption is due to the fact that the regression models are only fitted after the tree has been grown.

A new paper [1] has aimed to tackle this issue with a new split selection criteria that works under the assumption that linear regression models are applied in the two subsets corresponding to the two branches of a model tree and proposes an efficient algorithm to find the axis-parallel split that minimises the sum of squared errors.

This criteria has been implemented in M5' in WEKA [2]. The aim of this is to compare it to the existing split selection criteria in respect to tree size and predictive accuracy to assess the validity of this new criteria.

2 Method

The algorithm [1] addressed the selection split criteria issue with three main changes to the CART algorithm:

1. The leaves would return the mean of the corresponding training observations. This was changed to fitting a ridge regression in each leaf. That is, for a unit with features, x_{new}, that falls in a leaf S, the tree prediction is

$$\mu(x_{new}) := x_{new}^t (X_S^t X_S + \lambda)^{-1} X_S^t Y_S$$

where Y_S is the vector of y-values of the training observations that fall in leaf S, X_S is the corresponding design matrix for these observations, and $\lambda \epsilon R^+$ is a regularization parameter.

2. The new criteria must take into account that there is a linear regression model on the leaves. This is done by following a greedy strategy at each node

that finds the optimal splitting point. This means that for each candidate split which has two child nodes, the total Mean Squared Error (MSE) is minimized after fitting a penalized linear model on each child.

3. Finally, cross-validation stopping criteria is used to determine when to construct a leaf node as opposed to continue splitting. This has helped to find the optimal splitting point and therefore helps change 2 become computationally efficient. After the optimal split is selected, the R^2 improvement is introduced by the potential split calculated. If R^2 is increased by the potential split such that it is less than a predetermined percentage, a leaf node is created and the splitting is stopped. This has allowed decision trees to be created that have the ability to create large nodes with smoother aggregate functions. This is while it takes smaller nodes which mimic the performance of the classic CART algorithm.

3 Experimental results

To conduct this experiment, the new algorithm was implemented and compared to the original algorithm by testing on different datasets. The experiment was run using WEKA [2] to calculate the root relative squared error and the measure number of rules. These would allow us to see the predictive error of the algorithm and the number of leaf nodes.

Unfortunately, due to experimental restrictions of a slow laptop to run the experiments, not all datasets could be completed, namely datasets with over 2 million instances. Although netLibNative was used to try and speed up the experiment, the experiments could not be run in the allotted time for the assignment.

Some results of the experiment are shown in tables 1 and 2. (1) is the original algorithm and (2) is the newly implemented algorithm.

Table 1: Root Relative Squared Error (key

Dataset	**(1)**	**(2)**	
2dplanes	22.70	22.70	
abalone	65.95	67.08	
autoMpg	34.86	37.72	○
auto-price	37.79	44.77	○
cal-housing	48.50	44.53	●
cpu-act	14.79	14.53	
cpu-small	17.32	17.92	
delta-ailerons	54.35	55.72	○
delta-elevators	60.04	59.66	●
diabetes-numeric	91.71	93.80	
elevators	32.31	34.87	○
housing	40.83	41.94	
kin8nm	60.82	49.76	●
pol	15.61	17.11	
puma8NH	56.91	57.05	
pyrim	66.96	74.12	
servo	35.85	34.61	
stock	14.35	13.80	
triazines	87.09	86.41	
wisconsin	98.53	98.68	

○, ● statistically significant improvement or degradation

Table 2: Measure Number of Rules

Dataset	**(1)**	**(2)**	
2dplanes	2.00	2.00	
abalone	8.59	10.10	
autoMpg	3.90	2.67	
cal-housing	206.77	210.25	
auto-price	7.69	2.91	●
cpu-act	48.57	9.93	●
cpu-small	51.13	22.76	●
delta-ailerons	24.55	10.29	●
delta-elevators	7.21	6.22	
diabetes-numeric	1.65	1.98	
elevators	34.54	39.00	
housing	12.96	5.28	●
kin8nm	109.69	85.30	●
pol	170.47	102.93	●
puma8NH	28.03	16.79	●
pyrim	2.26	1.40	
servo	6.06	3.66	●
stock	44.68	18.52	●
triazines	3.65	2.56	●
wisconsin	3.15	1.62	

○, ● statistically significant improvement or degradation

4 Evaluation

Table 1 shows the prediction accuracy for each algorithm. In the majority of datasets, the accuracy is very similar. In the 20 datasets, 13 of them had no statistically significant difference at 0.05 significance. Only cal-housing, data-elevators and kin8nm had a significant degradation, but in most cases it was not a large margin. In datasets autoMpg, auto-price, delta-ailerons and elevators there was a significant improvement. Overall, the predictive accuracy did not change significantly in most datasets and was significant improvement more than a significant degradation in the others.

Table 2 shows the size of the decision tree for each algorithm. In 11 of the 20 datasets, there was a significant degradation of leaves, which means there was less linear regression models fitted at the root nodes. This shows that the tree size of the new algorithm is significantly smaller for most datasets experimented on.

5 Conclusions

To conclude, overall there was no statistically significant change in the predictive accuracy for most datasets, but there was a significant decrease in tree size. When choosing between two algorithms, there needs to be a compromise between complexity and predictive accuracy. Predictive accuracy between the two algorithms are similar, but the complexity of the new algorithm is significantly less than the original. Occam's Razor theory states 'the best theory is the smallest one that describes all the facts' allows us to conclude that the less complex algorithm is a better choice.

References

[1] Sören R. Künzel, Theo F. Saarinen, Edward W. Liu, and Jasjeet S. Sekhon. *Linear aggregation in tree-based estimators.* Journal of Computational and Graphical, 0(0), 2022.

[2] Ian H. Witten, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, Burlington, MA, 3 edition, 2011.