JHU AAP EN.605.649.83.FA21 Project 3 Report:

Decision Trees – ID3 Classification & CART Regression

Mauro Chavez Mauro.antoine.chavez@gmail.com | MCHAVEZ8@JH.EDU
AAP M.S. Bioinformatics Student *(858)354-7465*
San Diego, CA

Abstract

This document summarizes the third project deliverable for Introduction to Machine Learning taken through Johns Hopkins AAP in Fall of 2021. This project involved implementing two types of decision trees, CART for regression tasks and ID3 for classification tasks. The ID3 tree chooses attributes to split on for each node using the calculation of information gain while the CART tree calculated resulting mean squared error (MSE) for partitions made by each split. Each tree included methods for controlling its size. The ID3 tree had a pruning procedure to iteratively remove children of nodes when those children did not significantly improve classification accuracy. The CART tree included an option to stop growing the tree once MSE in a partition fell below a certain threshold. For the classification tree, numeric attributes had to be transformed prior to learning. One allowance during this assignment was that features that functioned as unique identifiers could be removed prior to learning. The UCI Machine Learning Repository data sets for Breast Cancer, Car Evaluation, and Congressional Vote, were used for classification and the Abalone, Computer Hardware, and Forest fires data sets were used for regression.

Video demonstration of code can be found here: https://youtu.be/O3SrcTdobWo

1 Introduction

This assignment required implementing decision trees. The ID3 approach would be taken for classification tasks and the CART approach would be followed for regression tasks. Both these trees incorporated some mechanism to control their growth. The CART tree was able to control its growth as it was built while the ID3 tree required running tests on a completed tree to see how it might be altered to decrease size while maintain performance. Similar to past assignments, the Breast Cancer, Car Evaluation, and Congressional votes data sets were used for classification learning tasks while the Abalone, Computer Hardware, and Forest fire data sets were used for used for regression learning tasks. Both are available from the UCI machine learning repository. An important consideration for this assignment was runtime. Many steps to the ID3 and CART algorithms required profiling partitions of the data for different statistics. Good design would allow a single pass of a data set to compute each partitions profile such that for subsequent steps the same data did not have to be passed over again for recalculation.

1.1 Problem Statement

The high-level goals of this assignment were to implement two decision trees. The first being an ID3 classification tree that calculated gain-ratio as the splitting criteria. Additionally, ID3 trees had to support reduced error pruning to cut down on tree size. Similarly, CART decision trees were used for regression. When initialized with a maximum partition mean squared error threshold, construction of the tree could be exited early. As the tree is built, if a partition is created where the mean squared error of data points in that partition falls below the threshold, growth of the tree is stopped along that branch. Both the ID3 and CART trees were to be tested on three data sets. For ID3 trees, the pruned and unpruned trees would be compared in classification accuracy and tree size. For CART trees, a hyperparameter tuning set would be used to compute a reasonable partition mean squared error allowance. Once this was computed, regression tasks would be completed for three other data sets. Testing of both trees required 5-fold cross validation for statistical comparison of results. For the hyperparameter tuning set and ID3 tree pruning set, 20% of the data points were set aside prior to learning.

1.2 Hypothesis

For this assignment, I anticipate performing better at classification tasks than regression ones. This is on account of the ID3 tree more discreetly splitting the data than the CART tree. Given the two algorithms, the ID3 tree is able to split exactly on feature value categories while the CART tree must make generalization about the resulting partitions. In short, the ID3 approach sounds more powerful to me with the admission that it's task is simpler. Furthermore, past assignments have shown more difficulty with regression tasks compared to classification ones. When it comes to pruning of the ID3 tree, I anticipate modest trimming of ~20-45% of nodes with a resulting decrease of ~5-15% classification accuracy. This is on account of my analysis utilizing curated data sets that I assume have had redundant examples edited from them. I am guessing that when the tree is built the first time from the data, that it will be closer to optimal than if we were working with a less standardized data set. For tuning of partition MSE allowance, modest ones will outperform smaller or larger ones. MSE's calculated using 10-15% error on class average should give enough wiggle room to help classification but not so much as to over generalize.

2 Data Pre-Processing

An important step to the data processing was to drop attributes that would function as identifiers of the data points. Given no restrictions, these would result in a tree where the identifier goes on to assign class values. This decision tree would memorize the data based off of data point identifiers. To prevent identifiers from defining illegitimate separators of the data, they were removed. For regression activities, no new transformations were added to the protocol from past assignments. In order to break the input data into folds, the class values were first discretized into 100 equal width bins. This step was required in order to define folds with comparatively equal class distribution.

For classification activities, numerical values had to be broken down into categories as these categories would group data points into partitions for the calculation of entropy and information gain. The procedure to do this involved iterating through the columns of the input data. For each numerical column, the examples were grouped according to class value. From there, the average feature value for each class was computed. Then, the classes were ranked in increasing average class value. For each pair of classes, the midpoint was calculated. The midpoints between the average feature value for each class were then used to define the edges of bins that were used to categorize the data. Numerical values in the data were then replaced with "vals_lt_X_gt_Y" where X defined the right edge of the bin (bin contains values less than X) and Y defined the left edge of the bin (bin contains values greater than Y).

2.1 Abalone Data

Abalone data was used for regression where attributes were used to predict age. Age appeared in the data set as the rings column as ring counting is used as a proxy for age. The "sex" attribute of this data set was broken out as nominal data resulting in sex_m and a sex_f columns.

2.2 Breast Cancer Data

Breast cancer data was used for classification where attributes were used to predict cancer class. Cancer class for this sample was either 2 for benign or 4 for malignant. There existed some "?" values in this data set. These were replaced with null values and imputed using the column mean. The sample column was dropped as data points were instead identified by indices in the data frame.

The following columns were categorized using the procedure described in section 2: clump_thickness, uniformity_of_cell_size, uniformity_of_cell_shape, marginal_adhesion, single_epithelial_cell_size, bare_nuclei, bland_chromatin, normal_nucleoli, mitoses. For details such as average class feature value and the resulting categories, please see `/command_output/ID3_decision_tree_car_[no/w]_prune.txt.`

2.3 Car Data

Car data was used for classification in order to predict acceptability. Similar to the first assignment and second, the buying, maint, safety, and acceptability attributes were normalized into ranks where 0 was the best/most desirable. Nominalization of ordinal columns with clear value rank was performed on features: doors, persons, and lug_boot attributes, as these were broken out into nominal data columns.

As described, due to some features being broken down into nominal columns, they were then considered for categorizing as they were read as numerical attributes. The following columns were categorized using the procedure described in section 2: buying, maint, safety, doors_2, doors_3, doors_4, doors_5more, persons_2, persons_4, persons_more, lug_boot_big, lug_boot_med, lug_boot_small. For details such as average class feature value and the resulting categories, please see `/command_output/ID3_decision_tree_car_[no/w]_prune.txt.`

2.4 Forest Fire Data

Forest fire data used for regression in order to predict burnt area. To start, the month and day attributes were broken down into nominal data columns. It was recommended in the data description file for this set that the area be log transformed as values are heavily skewed towards 0.00. However, when $\log_{10}$ transforming this column, all 0.00 values would be replaced with -inf in the data frame which was not useful for regression. To correct for this, a noise factor of 0.0000001 was added to each data point's area attribute. After this adjustment, log transformation gave non infinite values to each data point. This approach had mixed results in assignment two but given the increased sophistication of the CART algorithm, the same approach as utilized here.

2.5 House Votes Data

The House votes data set was used for classification where voting history was used to predict whether there would be a yes vote on the crime bill. Every attribute was broken down into nominal data columns. Similarly, to the car data, when breaking these attributes into nominal columns, they were read as numerical attributes. Instead of categorizing them as was the procedure for the car attributes, these were instead re-casted as strings. Each vote was broken down into nominal columns, but these Boolean features were read as strings not numbers, so no categorization occurred.

2.6 Computer Hardware (aka Machine) Data

The computer hardware data set was used for regression where the estimated relative performance was used as the target of prediction. The vendor and model_name attributes were dropped as they did more to serve as data point naming than actual device performance quantification.

3 Experimental Approach

3.1 ID3 decision tree classification

This decision tree would handle classification tasks and use gain ratio to determine the splitting criteria at a node. Classification would be ran on each data set to build and asses bot pruned and unpruned trees. The ID3 tree would be ran on the Breast Cancer, Car Evaluation, and Congressional Votes data sets undergoing 5-fold cross validation. In the case that learning included a reduced error pruning step, 20% of the data had to be partitioned out ahead of 5-fold cross validation. A tree would be built for each fold where folds contained 80% of the input data. Each tree would then use the 20% saved as a validation set to prune the tree. Pruning involved iterating through each node of the tree and removing it's children. After removing it's children each data point in the validation set would be classified. If classification accuracy improved or stayed the same, the children were omitted. This was repeated until a check of every node in the tree was unable to improve accuracy or decrease tree size. If no pruning step occurred, 100% of the data was used for defining folds used to build and test the tree. The ID3 tree accuracy was tested using both the pruned and unpruned trees for each data set utilizing 5-fold cross validation. Accuracy was measured by recording the classification accuracy of each fold which is defined as the percentage of test examples classified correctly. Additionally, the total size of the tree for both pruned and unpruned trees was recorded to understand the effects of pruning.

3.2 CART decision tree regression

The CART regression tree used mean squared error to determine which attribute to split on at each decision point. To control tree size, the CART tree accepted an optional mean squared error threshold parameter where once a partition had mean squared error less than this value, these partitions were classified as leaf nodes and used for prediction. Setting the partition mean squared error allowance to zero would result in no early stopping and no control over the growth of the tree. Similar to the ID3 tree, 5-fold cross validation was performed for the Abalone, Computer Hardware, and Forest fire data sets. Additionally, 20% of the data was set aside prior to learning to test different partition mean squared error parameters. This procedure broke the 20% of input data into 5-folds. For each fold, the average class value in the training set was calculated and scaled by 0.00, 0.05, 0.1, 0.15, 0.2, and 0.25. The scaled class means were then squared. This provided a set of 6 mean squared error allowances to test on each fold of the data. For each allowance, the fold's training set was used to construct a tree and the test set was used to calculate mean squared error on test examples. Once each allowance had been tested using single fold, the mean squared error allowance with the minimum resulting mean squared error on the test set was saved. This was repeated for each fold. At the end, the best MSE allowance for each of the 5 folds was averaged. This provided the maximum partition MSE allowance to use in classifying the other 80% of the input data. Using the assigned MSE allowance, the 80% of remaining data was used for 5-fold cross validation. For each fold, a tree was constructed using the training set and assigned MSE allowance.

The resulting tree was used to predict class values for the test set. The resulting mean squared error on the test set and tree node count were recorded and averaged across each fold for each data set.

4 Experimental Results

4.1 ID3 Classification Results with and without Pruning

			Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
Breast Cancer (~700 dp)	No Pruning	TNC	77	71	103	93	101	89.0
		CA	0.8936	0.9071	0.9643	0.9281	0.9856	0.93574
	With Pruning	TNC	11	21	13	17	29	18.2
		CA	0.9381	0.9464	0.9640	0.9820	0.9910	0.9643
Car Evaluation (~1700 dp)	No Pruning	TNC	203	218	228	196	174	203.8
		CA	0.841	0.7659	0.7803	0.6474	0.5756	0.72204
	With Pruning	TNC	29	38	24	32	24	29.4
		CA	0.7266	0.7374	0.8261	0.6764	0.7273	0.73876
Congressional (~450 dp)	No Pruning	TNC	5	5	5	5	5	5
		CA	1.0	1.0	1.0	1.0	1.0	1.0
	With Pruning	TNC	5	5	5	5	5	5
		CA	1.0	1.0	1.0	1.0	1.0	1.0

Table 1: Results of running ID3 classification on Breast Cancer, Car Evaluation, and Congressional Vote data sets across 5-fold cross validation with and without pruning. TNC rows report Total Node Count of the resulting tree and CA rows show resulting Classification Accuracy on fold test set. Pruning involved removing 20% of the data prior to 5-fold cross validation to use as a pruning set for the tree built for each fold.

4.2 Classification MSE (mean squared error) cutoff tuning

	0.0: No Stopping		**0.05**		**0.1**		**0.15**		**0.20**		**0.25**	
	MSE allowance	Avrg MSE on test set	MSE allowance	Avrg MSE on test set	MSE allowance	Avrg MSE on test set	MSE allowance	Avrg MSE on test set	MSE allowance	Avrg MSE on test set	MSE allowance	Avrg MSE on test set
Abalone (~4180 dp)	0.0000	10.45	0.2467	10.31	0.9868	10.63	2.2203	9.19	3.9471	8.59	6.1674	9.06
Computer Hardware (~210 dp)	0.0000	21708	24.66	21700	98.67	21794	222	22784	394.66	21760	616.65	42991
Forest Fires (~520)	0.0000	28.35	0.0214	28.38	0.0856	28.45	0.1926	28.471	0.3424	27.59	0.5350	30.07

Table 2: Results of doing MSE cutoff tuning using hyperparameter validation set of 20% of the data removed prior to learning. Each MSE allowance calculated using a proportion of the mean class value (0.0, 0.05, 0.1, 0.15, 0.20 and 0.25). Each MSE allowance tested for each iteration of 5-fold cross validation partitions.

Computer Hardware data set suffered from being smaller. For example, when testing MSE cutoff of 0.000, the resulting MSEs across the 5 validation partitions were: 107979, 513, 23 and 25. However, this distribution of MSEs stayed consistent across testing of other MSE allowances across each partition for other data sets as well. Across the testing of each MSE allowance across each fold, the partition of the data for each fold seemed to have more impact on the resulting MSE on the fold's test set than the actual MSE allowance used to build a tree from the training set.

Data Set	**Average best MSE allowance from each fold…**
Abalone	3.6017501755426293
Computer Hardware	24.666193539525192
Forest Fires	0.0856072901208139

Table 3: Selected MSE thresholds for building regression trees for each data set. Thresholds were chosen by calculating an MSE threshold given 0.0, 0.05, 0.1, 0.15, 0.20 and 0.25 of the class label average value as error tolerance and testing each on each fold of 5-fold cross validation. For each fold, the best MSE threshold was selected and the best MSE threshold across each were averaged to determine the MSE allowance used in future learning.

4.3 CART Regression Results

		Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
Abalone (~4180 dp)	TNC	565	627	655	659	681	637.4
	MSE	7.3622	8.3919	7.6692	7.4295	7.6692	7.4295
Computer Hardware (~210 dp)	TNC	47	65	59	67	61	59.8
	MSE	864.420	102.282	131.667	118.703	46.6843	252.751
Forest Fires (~520)	TNC	383	385	399	361	389	383.4
	MSE	30.423	23.085	35.344	38.942	35.906	32.740

Table 4: Results of running CART regression on Breast Cancer, Car Evaluation, and Congressional Vote data sets across 5-fold cross validation. TNC rows report Total Node Count of the resulting tree and MSE rows show resulting Mean Squared Error on fold test set. Optimizing of partition MSE allowance involved removing 20% of the data prior to 5-fold cross.

For better comparison between classification and regression, classification accuracy was computed allowing for 15% error between class label and prediction. The Abalone data set resulted in 60.88% of example classified correctly under this criterion. The computer hardware data set resulted in 95.65% of examples classified correctly. Unfortunately, due to the corrections enacted on the forest fire labels, 0.0% were classified correctly when only allowing for 15% error.

5 Discussion

With respect to classification tasks, the breast cancer and congressional data sets were well classified using the ID3 tree with 96% and 100% of example classified correctly respectively. On the other hand, the car evaluation data set did not do so well, sitting at ~74% of example classified correctly. This could be due to the data transformation of making nominal columns and then categorizing them. Procedurally, the information about each example should remain the same for the most part. However, examples where their feature value clearly matched the expected feature value of a different class were likely muddied in the transformation. This may explain the lower classification accuracy for the car evaluation data. That being said, the procedure of transforming numerical columns into categories as discussed in section 3.1 seemed successful for the breast cancer data set. The difference being that these numerical columns were not strings transformed into integer rankings. The interesting take away from classification is that the pruned trees dramatically cut the total tree size and did not hurt classification accuracy. For the breast cancer and car evaluation data sets, 80% to 90% of nodes were removed by pruning. In both cases, the classification accuracy rose. The congressional data set being much easier to split, didn't benefit from pruning especially because it already had a classification accuracy of 100%. This is an interesting result that I did not anticipate in my hypothesis.

The tuning of CART tree partition mean squared error allowance did not produce interesting results. This is mainly driven by the biggest determinant of resulting average MSE change being utilizing a different fold and not a different MSE parameter. That being said, following the procedure discussed in section 3.2, the resulting MSE allowances used in subsequent training did fall in the 0.05% to 0.01% of average class value squared range.

Similar to past assignments, the Forest Fires data set proved challenging to classify. Given that the range of class values goes from -7 to 3 as a result of the data transformation, assigning meaningful values doesn't seem possible. This is puzzling as the log transformation step was recommended by the data source. When conceptualizing regression performance as the % of examples with prediction within 15% of its actual value, classifying computer hardware performance was 95.65% effective. On the other hand, classifying abalone was only 60.88%.

Conclusion

I ultimately had more success with classification than regression. This assignment equipped us with algorithms that performed better than previous ones when considering both their runtime and accuracy. My two major take always are that my approach to tuning mean squared error allowance had mixed results and I may have sabotaged my ability to classify the Car Evaluation data set by over transforming the data. Furthermore, the congressional data set had the least amount of modifications and this is where decision trees performed the best.

I was very conscious about runtime and luckily my implementation is pretty quick. This is a result of me sacrificing memory for speed. When I iterate over a data frame to compute statistics used in splitting, I construct auxiliary data structures to index different components of each calculation. This meant that I often stored statistics of sub partitions in memory. To compute things relevant to a larger partition, I combined results of the sub partitions. When a split happened, I already had the information I needed about the partition. By adding data structures to store results of my calculations I was able to cut down on the amount of times I had to iterate through training examples when building the tree.

One complication to the regression tasks arose when processing the forest fire data. There were a few cases where there were two data points with identical feature values but different classes. For example, there were two fires at (x,y) = (8, 6). When log transformed using the procedure described above, one had area 0.52138 and the other had area -7.00000. This was an issue as the mean squared error of these two data points could never fall below the threshold set in building the regression tree. This resulted in an infinite loop of attempting to split these two data points with no success. The issue I ran into is one of data non-separability. I could opt to remove these data points but as this assignment utilized curated data sets, I didn't want to omit examples. Instead I added a condition to my build tree function that checked how many columns contained more than 1 value. If every column except for the class column had the same value spanning every example in the data frame, tree building was stopped. In these cases, have constructed trees that break the partition mean squared error allowance for these nodes that contain data points with identical feature values but different class assignments.