Introduction to ML - Decision Tree Coursework Report

rh4618, kd120, ad5518, prm2418

November 5, 2020

1 Implementation

Each of our decision trees is constructed using two classes inheriting from a `Node` class - a `LeafNode` containing a label, and a `TreeNode`, containing the attribute index to split at, the splitting value and the left and right subtrees. The decision trees are constructed by first ordering the given training set by value for each attribute, then finding an optimal split point for said attribute. This is determined to be the split by which we obtain the maximum information gain from respective labels in the two resultant subsets. The optimal split for the node is is determined to be the attribute split configuration with the highest information gain. When we are left with a dataset with all labels the same, we stop and generate a `LeafNode` with said label. Otherwise, we attempt to optimally split the current dataset with a `TreeNode` and repeat the process on the two resultant subsets.

To perform cross-validation, we first shuffle the given dataset, before splitting the dataset to ten equal-sized folds. We then iterate through the folds, with one being designated test set for each iteration, the remaining folds are concatenated into a single training set. We train a decision tree on each of the configurations of the data set and generate a confusion matrix for each. We obtain an average confusion matrix through the micro-averaging of these ten confusion matrices, from which our precision, recall and F1-measures are calculated for each class. The classification rate is calculated as the accuracy of the average confusion matrix.

The cross-validation method we use for pruning operations includes some additional steps, which we will talk about in section 4.2.

2 Evaluation & Results

We performed the evaluation separately on the clean and noisy dataset. An average confusion matrix was computed in each case via micro-averaging of all of the data points in each confusion matrix produced under each fold of cross-validation.

Using the confusion matrix, we computed a series of performance metrics for each class — precision, recall, and F1-score. These are stated in the classification reports below, along with the class-level averages (macro-averaging) for each of those metrics. Data in the tables have been rounded to five decimal places.

2.1 Evaluation on the clean dataset

From table 2, we can conclude that the model classifies the rooms with rather high overall accuracy, given the 0.972 average classification rate. It is also worth noting that the values of precision, recall and F1-score for all classes are at least above 0.95, signifying that all of the individual rooms are recognized accurately. From the column-wise comparison of different classes, it is apparent that Room 4 with the highest score in all of precision, recall and F1-score

is the room that is identified the best, while Room 3 with the lowest class-specific scores is the poorest. This result can also be verified by examining the confusion matrix: along the diagonal Room 4 has the highest value of 49.4 and, expanding along the row and column, the lowest values of False-Positives (FP) and False-Negatives (FN), while for Room 3 it is the complete opposite. The most common confusions seem to be between the rooms 2 and 3.

Average Confusion Matrix				
Actual / Predicted	Room 1	Room 2	Room 3	Room 4
Room 1	49.2	0	0.4	0.5
Room 2	0	48.1	1.6	0
Room 3	0.3	1.9	47.7	0.1
Room 4	0.5	0	0.3	49.4

Table 1: Average confusion matrix for 10-folds cross-validation on the clean dataset.

Classification Report				
Class	Precision	Recall	F1-score	Average classification rate
Room 1	0.98204	0.984	0.98302	0.972
Room 2	0.96781	0.962	0.96490	
Room 3	0.954	0.954	0.954	
Room 4	0.98406	0.988	0.98603	
Macro-averaging	0.97197	0.972	0.97198	

Table 2: Classification report after cross-validation on the clean dataset.

2.2 Evaluation on the noisy dataset

The same 10-folds cross-validation procedure was performed on the noisy dataset. Compared to the matrix obtained when evaluating on the clean dataset, the diagonal values of the resulting confusion matrix (table 3) are substantially smaller. Subsequently, all the False-Positives and False-Negatives counts have increased. A consequence of this observation is directly visible in the average classification rate in table 4, where the overall accuracy of the model has dropped from 0.972 to 0.8085, a near 17% decrease. This indicates the decision tree performs considerably worse when confronted with a noisy dataset.

From table 4, we see that the Room 4 again has the highest F1-score of approximately 0.8235. In contrast, the F1-score for the Room 1 class is the lowest. This corresponds with the relatively high number of confusions of Room 1 for different rooms also resulting in low recall for Room 1. Similarly to the result on the clean dataset, the confusions between the rooms 2 and 3 also seem to be relatively common.

Average Confusion Matrix				
Actual / Predicted	Room 1	Room 2	Room 3	Room 4
Room 1	38	2.9	2.3	3
Room 2	3.1	39.7	3.8	2.3
Room 3	3.6	4.6	42	2.5
Room 4	4.3	2.5	3.4	42

Table 3: Average confusion matrix for 10-folds cross-validation on the noisy dataset.

Classification Report				
Class	Precision	Recall	F1-score	Average classification rate
Room 1	0.82251	0.77551	0.79832	0.8085
Room 2	0.81186	0.79879	0.80527	
Room 3	0.79696	0.81553	0.80614	
Room 4	0.80460	0.84337	0.82353	
Macro-averaging	0.80898	0.80830	0.80832	

Table 4: Classification report after cross-validation on the noisy dataset.

2.3 Evaluation after pruning

The tables below list the key performance metrics for the pruned trees applied both to the clean and nosy datasets. In order to allow for clear comparison, we also repeat the previously stated results for the unpruned trees. All the numbers in the tables have been rounded to 5 decimal places. Table 5 details the performance comparison for the clean dataset, while table 6 covers that for the noisy dataset.

Before Pruning				
Class	Precision	Recall	F1-score	Average classification rate
Room 1	0.98204	0.984	0.98302	0.972
Room 2	0.96781	0.962	0.96490	
Room 3	0.954	0.954	0.954	
Room 4	0.98406	0.988	0.98603	
Macro-averaging	0.97197	0.972	0.97198	
After Pruning				
Class	Precision	Recall	F1-score	Average classification rate
Room 1	0.97747	0.99311	0.98523	0.96361
Room 2	0.95131	0.94644	0.94887	
Room 3	0.9347	0.932	0.93335	
Room 4	0.99082	0.98289	0.98684	
Macro-averaging	0.96358	0.96361	0.96357	

Table 5: Comparison of performance metrics before and after pruning, on the clean dataset.

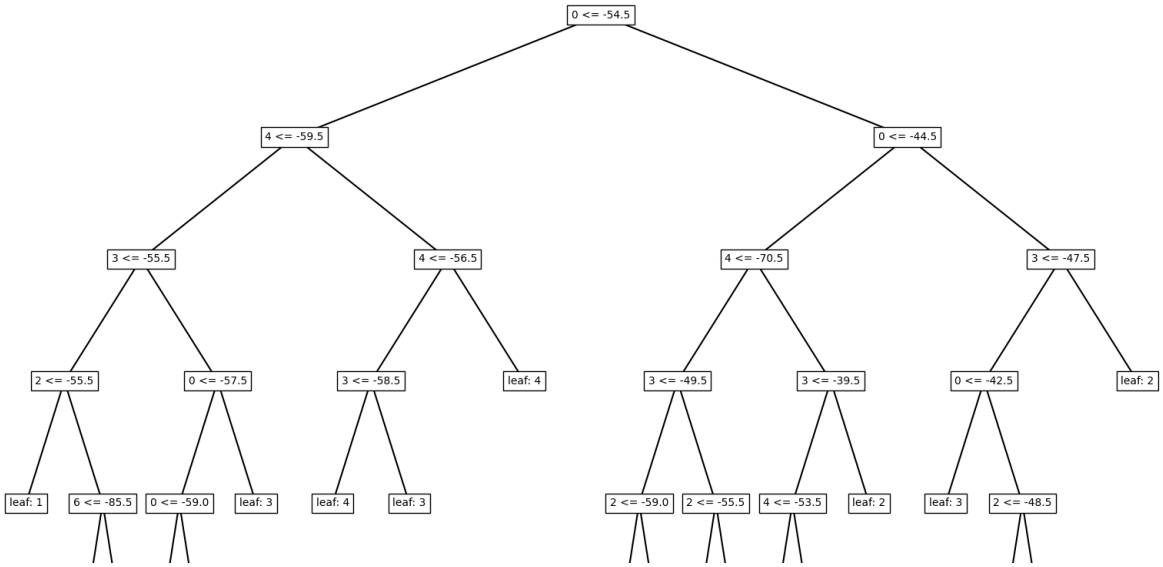
Before Pruning				
Class	Precision	Recall	F1-score	Average classification rate
Room 1	0.82251	0.77551	0.79832	0.8085
Room 2	0.81186	0.79879	0.80527	
Room 3	0.79696	0.81553	0.80614	
Room 4	0.80460	0.84337	0.82353	
Macro-averaging	0.80898	0.80830	0.80832	
After Pruning				
Class	Precision	Recall	F1-score	Average classification rate
Room 1	0.86922	0.89524	0.88204	0.87506
Room 2	0.88107	0.86452	0.87271	
Room 3	0.86052	0.85588	0.85819	
Room 4	0.89011	0.88554	0.88782	
Macro-averaging	0.87523	0.87530	0.87519	

Table 6: Comparison of performance metrics before and after pruning, on the noisy dataset.

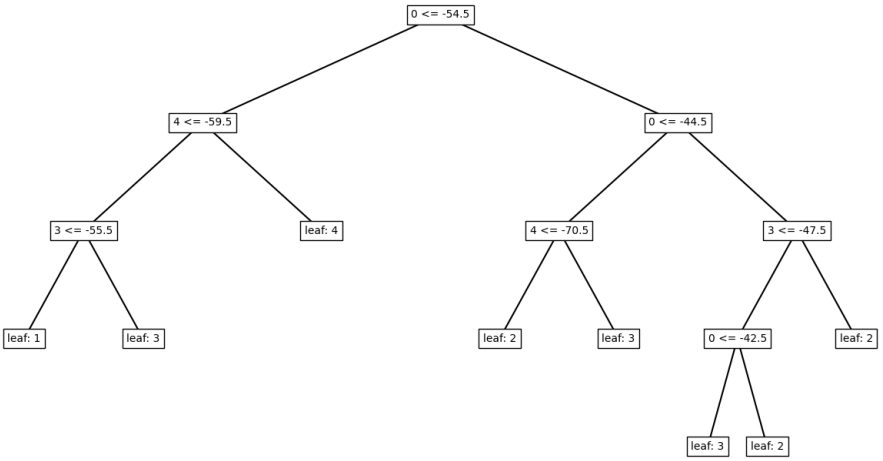
For the clean dataset, from table 5, it is observed that pruning actually reduces the overall accuracy by less than 1%. However, on the noisy dataset, the pruning does manage to boost the accuracy by roughly 7%. A similar result can be reported for the macro-averaged F1-scores (near 1% decrease for clean dataset and about 7% increase for noisy dataset), consistent with that revealed by simply examining the overall accuracy. Therefore, the overall effect of pruning seems to be highly positive for the performance of trees applied to the noisy dataset while perhaps being slightly negative for the trees applied to the clean datasets. The latter finding can possibly be attributed to the relatively low number of samples in the validation dataset and the resulting imprecision of the pruning process.

3 Diagrams

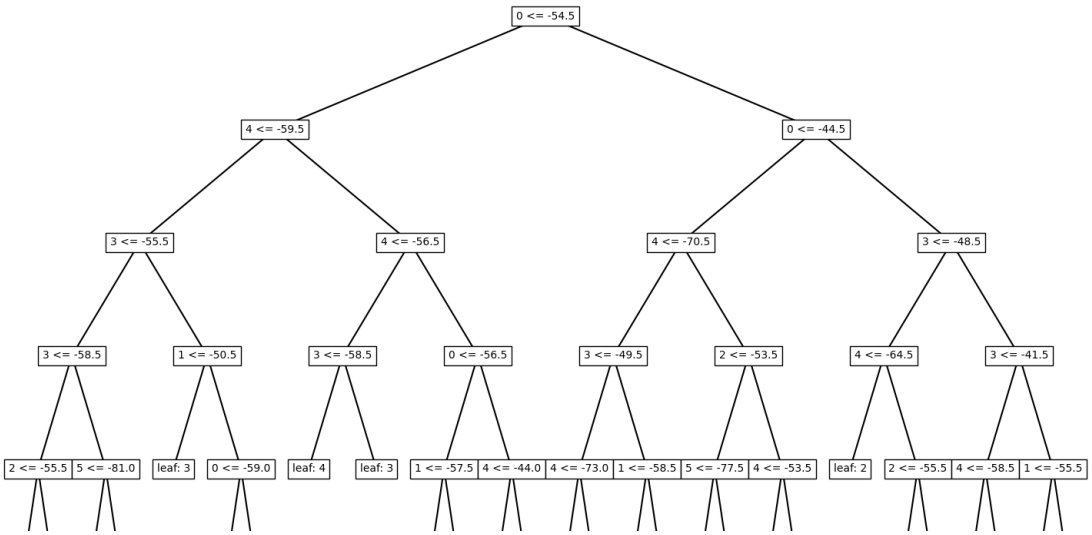
3.1 Clean Dataset Tree (Cropped)



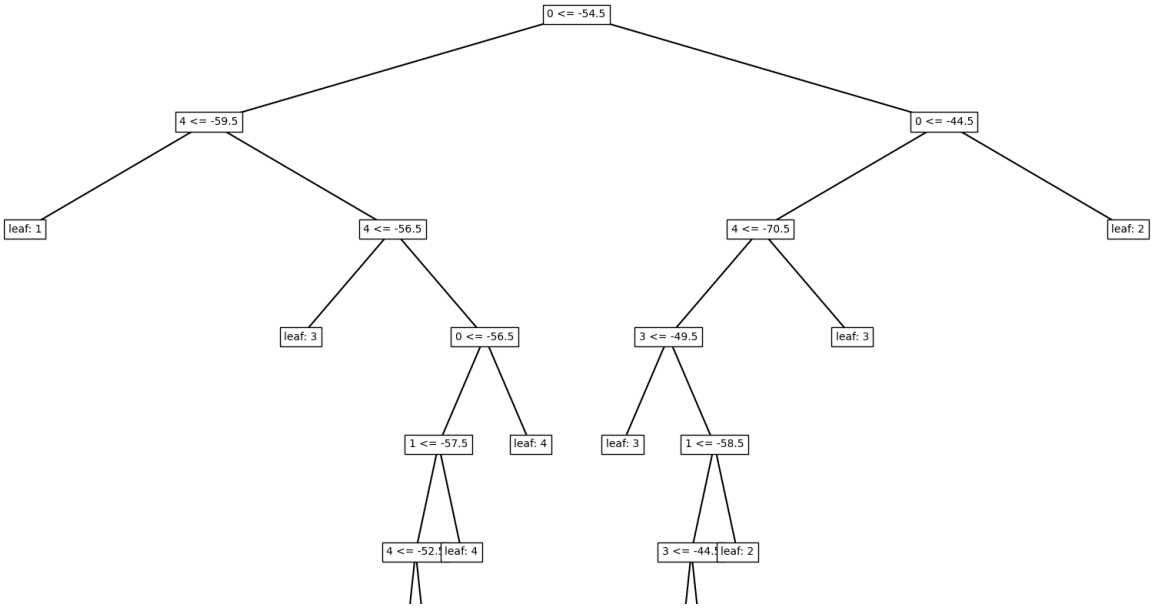
3.2 Clean Dataset Tree after Pruning



3.3 Noisy Dataset Tree (Cropped)



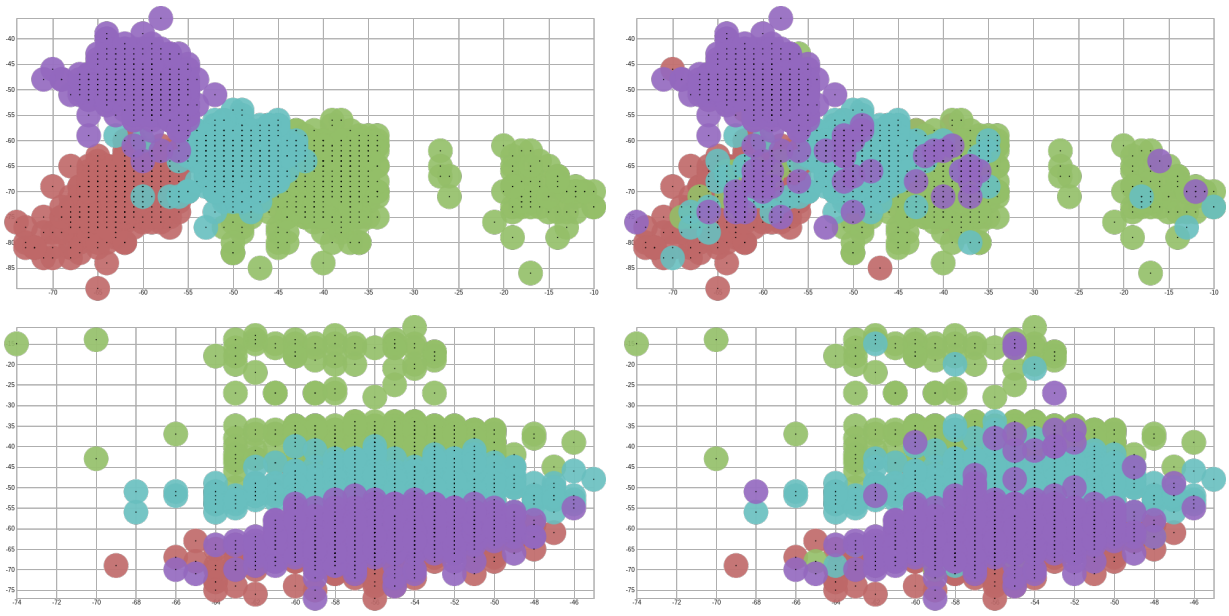
3.4 Noisy Dataset Tree after Pruning (Cropped)



4 Questions and Answers

4.1 Noisy-Clean Datasets Question

As can be seen from the tables 5 and 6 above, the performance on the clean dataset is considerably better than the performance on the noisy dataset in all of the used metrics, including precisions, recalls, F1-scores and the overall classification rate. These results can be attributed to the differences in the data sets.



The graphs above show the data points from the clean (on the left) and noisy (on the right) datasets placed according to the signal strength from the emitters 1 and 5 (upper) and 2 and 4 (lower). Even though the plots are unable to express all dimensions of the underlying data, it is apparent that the boundaries between the classes are much clearer in the clean dataset than in the noisy one.

The unclear boundaries between the data in the noisy dataset pose a challenge for the decision trees training algorithm that attempts to construct a model fitting every point in the training data. As the underlying data are noisy, this results in overfitting and a highly complex decision tree with overly deep branches. Such a decision tree is then unable to generalise well to provide reliable predictions on previously unseen test data. Pruning on the validation data can partially mitigate this issue by removing some of the superfluous branches.

However, the difference in performance is significant even when the evaluation is done on the pruned decision trees. This difference can be partially attributed to the misclassification of the noisy data points with imprecise attribute values or wrong labels that are present in the testing data. Additionally, as the validation data used to prune the trees are also noisy, the pruning process is inherently imprecise and may potentially result both in incorrect removal of branches significant for correct classification as well as leaving some of the branches that are unnecessary and result in overfitting model.

On both clean and noisy datasets, the performance of the unpruned decision trees appears to be the best when classifying Room 4, as this class has the highest F1-score for both trees. However, there is a slight difference in the relative performance on the Room 1 class between the two sets. While the F1-score for Room 1 is the second-highest when the clean data set is used, it is the lowest when using the noisy set. This might suggest that there is more noise in the data for Room 1, causing a decrease in performance.

4.2 Pruning Question

For our pruning experiment, we perform cross-validation in a similar manner to our standard cross-validation method described above. However, due to our requirement of a validation dataset, we perform cross-validation on what previously was the training set to obtain nine different configurations of validation and training datasets. In each of these configurations, the

validation dataset consists of one fold of the original data, while the training set contains eight such folds. After performing the additional splitting of the data, we train the decision tree on the training data and then prune it using the validation set. We then generate and average the confusion matrices as per above. In total, for 10 folds, 90 decision trees are trained and pruned with 90 corresponding confusion matrices generated in the process.

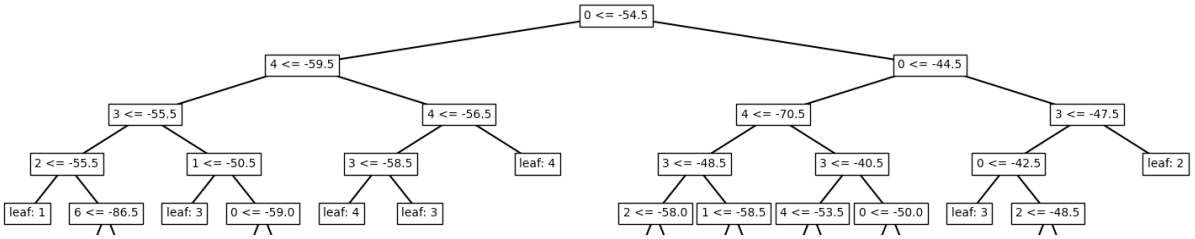


Figure 4.2.1: Example of decision tree trained on the clean data set before pruning (Not all of the tree is displayed)

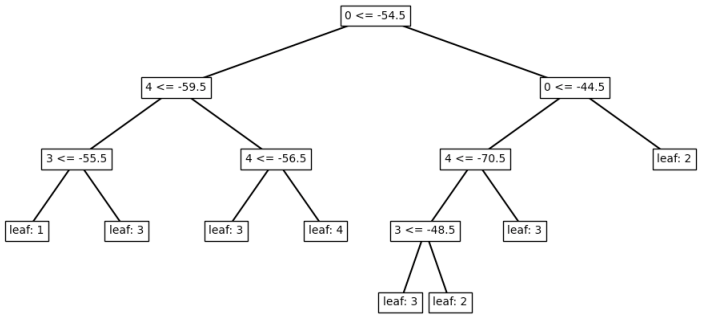


Figure 4.2.2: Same decision tree after pruning

To prune an individual tree, we take a bottom-up approach, where we begin at the lowest tree nodes that are connected solely to two leaf nodes. We determine the majority label from the children of the current node and check the hypothetical number of successes if we were to replace it with a leaf node of said majority label. If a leaf node results in a greater or equal classification rate, we replace the current node. This process is recursively propagated up the tree until either no nodes are available for pruning, or could be pruned to obtain a higher classification rate on the validation set. The resultant tree is then evaluated on the test set.

The specific results after pruning are listed in the results section above. The main observation is that there was a relatively small reduction in the average classification rate of our decision trees on clean data sets, but a significant improvement to the performance, in all precision, recall and F1-measure metrics on the noisy data set, an improvement of approximately 7% to the average classification rate.

This improvement to the average classification rate on the application on the noisy dataset seems emblematic of the effects of reducing the complexity of the tree trained on the training set, suggesting that previously there was a degree of overfitting. The reduction in complexity is evident when the tree is visualised.

4.3 Depth Question

As seen in Table 7, the maximal depths of the trees generated from the clean dataset (before and after pruning) were lower than the depths of the trees generated from the noisy dataset.

Dataset type	Maximal depth	Prediction accuracy
Clean dataset	15	0.97
Clean dataset after pruning	5	0.96
Noisy dataset	20	0.81
Noisy dataset after pruning	8	0.87

Table 7: Maximal tree depth and prediction accuracy for different datasets

This is likely because the trees from the noisy dataset are over-fitting the data and therefore have a higher complexity. The depths of the pruned trees were lower than the base trees for both datasets, which is expected since pruning reduces the complexity.

The relationship between the maximal depth and the percentage accuracy of decision trees could be explained using bias-variance tradeoff. A very large maximal depth would mean that a generated decision tree can have a large complexity. Such a tree would likely have a high variance and would be over-fitting the data, which in turn would reduce its prediction accuracy. A very low maximal depth would have the opposite effect - it would result in a highly biased tree with low complexity, which again would result in reduced accuracy due to the under-fitting of the data.

The pruning of the tree generated by the clean dataset resulted in a slight decrease in prediction accuracy (1%) and a large decrease in maximal depth (over 60%). Given the still large percentage accuracy (96%), this could be considered as an improvement over the base decision tree, which had far higher complexity. Given the slight decrease in accuracy, it could be speculated that the pruned tree was more biased than the original tree and slightly under-fit the data.

The pruning of the tree generated by the noisy dataset resulted in a large increase in prediction accuracy (6%) and a large decrease in maximal depth (over 50%) - this could be considered a larger improvement compared to the result for the clean dataset. The pruning resulted in a significant complexity reduction, (which in turn reduced the maximal depth) and also a significant accuracy increase - this suggests that the original decision tree was over-fitting the data and had a high variance.