Imperial College London

Imperial College London

Department of Computing

Introduction To Machine Learning - Decision Tree Coursework

Authors:
Sherif Agbabiaka, CID: 01865621
Dominic Justice Konec, CID: 01945883
James Nock, CID: 01502007
Louise Davis, CID: 01909253

3 November 2022

Contents

1 Visualisation

Figure 1 shows a visualisation of a decision tree model that was trained on the clean dataset. The implementation for this is based on the Reingold-Tilford algorithm [1] [2] [3].

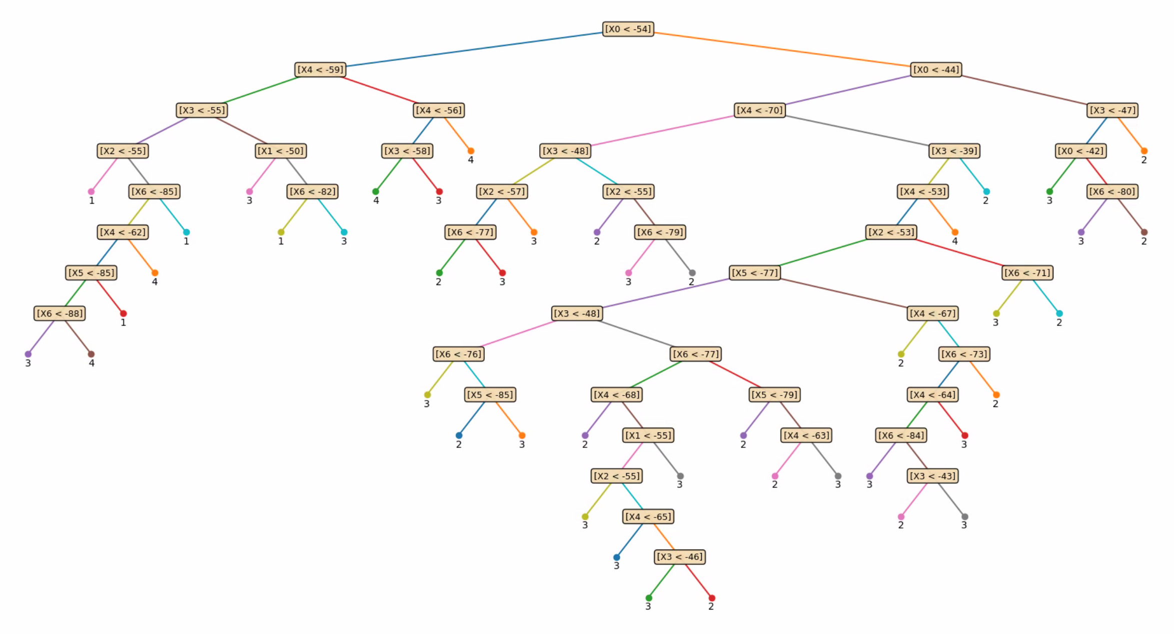

Figure 1: A decision tree generated using the clean dataset

2 Evaluation - Without Pruning

2.1 10-Fold Cross Validation Metrics

2.1.1 Confusion Matrix

$$
\begin{array}{cc}
& \text{Predicted Room} \\
& \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\
\text{Actual} & \begin{bmatrix} 492 & 0 & 2 & 6 \\ 0 & 480 & 20 & 0 \\ 2 & 20 & 477 & 1 \\ 3 & 0 & 1 & 496 \end{bmatrix} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array}
\end{array}
\qquad
\begin{array}{cc}
& \text{Predicted Room} \\
& \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\
\text{Actual} & \begin{bmatrix} 391 & 26 & 35 & 38 \\ 29 & 407 & 35 & 26 \\ 31 & 33 & 413 & 38 \\ 39 & 28 & 47 & 384 \end{bmatrix} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array}
\end{array}
$$

Table 1: Confusion matrices for the non-pruned clean and non-pruned noisy datasets, shown left and right respectively

2.1.2 Accuracy

Dataset	Non-Pruned Accuracy
Clean	0.9725
Noisy	0.7975

Table 2: The 10-fold cross validation non-pruned accuracy, derived from both the clean and noisy datasets

2.1.3 Recall and Precision

	Clean Dataset		Noisy Dataset	
Room	Precision	Recall	Precision	Recall
1	0.9899	0.9840	0.7980	0.7980
2	0.9600	0.9600	0.8239	0.8189
3	0.9540	0.9540	0.7792	0.8019
4	0.9861	0.9920	0.7901	0.7711

Table 3: The 10-fold cross validation precision and recall for each room, derived from both the clean and noisy datasets

2.1.4 F1 Measure

Room	Clean F1	Noisy F1
1	0.9870	0.7980
2	0.9600	0.8214
3	0.9540	0.7904
4	0.9890	0.7805

Table 4: The 10-fold cross validation F1 measure for each room, derived from both the clean and noisy datasets

2.2 Result Analysis

As seen in Table 4, Room 3 has the worst performance with clean data, whereas Room 4 has the best. Table 1 shows Rooms 2 and 4 are never mixed – understandably as they are not adjacent. Rooms 1 and 2 are also never misidentified as one another. The majority of mispredictions come from Rooms 2 and 3, with 20 of each being predicted as the other: likely due to the adjacency between the two rooms. With noisy data, Room 4 performs the worst and Room 2 the best.

2.3 Dataset Differences

As expected, the performance of the training algorithm is much better when tested using clean data as opposed to noisy data - evident from the consistently higher F1 scores in Table 4. This is likely due to over-fitting the noisy training-set; the built tree without pruning was much deeper than the clean tree – 19.4 nodes deep on average vs 12.3, seen in Table 9. This larger noisy depth comes from more attribute overlap between rooms, so more splits are used to make all leaves pure.

3 Pruning

3.1 10-Fold Cross Validation Metrics

3.1.1 Confusion Matrix

$$\begin{array}{cc} & \text{Predicted Room} \\ & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \text{Actual} & \begin{bmatrix} 499 & 0 & 1 & 0 \\ 0 & 479 & 21 & 0 \\ 8 & 17 & 474 & 1 \\ 5 & 0 & 1 & 494 \end{bmatrix} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array} \qquad \begin{array}{cc} & \text{Predicted Room} \\ & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \text{Actual} & \begin{bmatrix} 447 & 10 & 12 & 21 \\ 19 & 433 & 32 & 13 \\ 28 & 27 & 441 & 19 \\ 22 & 26 & 18 & 432 \end{bmatrix} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array}$$

Table 5: Confusion matrices for the pruned clean dataset and pruned noisy dataset, shown left and right respectively

3.1.2 Accuracy

Dataset	Non-Pruned Accuracy	Pruned Accuracy	Percentage Increase (4 s.f.)
Clean	0.9725	0.9730	0.05141
Noisy	0.7975	0.8765	9.906

Table 6: The 10-fold cross validation non-pruned and pruned accuracy, derived from both the clean and noisy datasets

3.1.3 Recall and Precision

	Clean Dataset		Noisy Dataset	
Room	Precision	Recall	Precision	Recall
1	0.9746	0.9980	0.8663	0.9122
2	0.9657	0.9580	0.8730	0.8712
3	0.9537	0.9480	0.8767	0.8563
4	0.9980	0.9880	0.8907	0.8675

Table 7: The 10-fold cross validation precision and recall for each room, derived from both the clean and noisy datasets after pruning

3.1.4 F1 Measure

Room	Clean F1	Noisy F1
1	0.9862	0.8887
2	0.9618	0.8721
3	0.9509	0.8664
4	0.9930	0.8789

Table 8: The 10-fold cross validation F1 measure for each room, derived from both the clean and noisy datasets after pruning

3.2 Result Analysis After Pruning

Table 6 shows a negligible percentage accuracy increase of 0.05% after pruning the clean dataset. This is expected, as in the clean dataset, often the original subtree was as good as the best pruning option. As such, no-pruning was preferred due to the low tree complexity [1]. The noisy dataset saw a large improvement of 9.9% – having all pure nodes in a noisy dataset model causes over-fitting, which pruning helps to solve. Aside, Room 3 performs poorly both with and without pruning.

3.3 Depth Analysis

	Clean Dataset			Noisy Dataset		
Depth	Non-pruned	Pruned w/o HP	Pruned w/ HP	Non-pruned	Pruned w/o HP	Pruned w/ HP
Max	12.30	9.300	12.40	19.40	9.700	11.00
Mean	6.978	5.245	6.898	11.02	5.733	6.717

Table 9: The 10-fold average mean and max depths of trained trees, before pruning and after pruning with and without using a hyperparameter (to 4 s.f.)

[1]This is controlled by a hyperparameter discussed in Section 3.3.

A hyperparameter (HP) was introduced to control the tendency of the algorithm to prune.[2] When always preferring pruning, the clean performance decreases: it causes the tree to under-fit the training set and Table 9 shows this via a reduction in the tree depth. Tuning this HP restores performance. Yet, the noisy data has much greater tree depth pre-pruning – suggesting over-fitting. This is reduced to a similar depth of the the clean tree post-pruning, with better performance.

Bibliography

[1] E. Reingold and J. Tilford, "Tidier drawings of trees," *IEEE Transactions on Software Engineering*, vol. SE-7, no. 2, pp. 223–228, Mar. 1981, ISSN: 0098-5589. DOI: `10.1109/TSE.1981.234519`. [Online]. Available: `http://ieeexplore.ieee.org/document/1702828/`.

[2] R. Lim. "Algorithm for drawing trees," Rachel Lim's Blog. (Apr. 20, 2014), [Online]. Available: `https://rachel53461.wordpress.com/2014/04/20/algorithm-for-drawing-trees/`.

[3] B. Mill. "Drawing presentable trees." (), [Online]. Available: `https://llimllib.github.io/pymag-trees/`.

[2]This parameter uses the tree complexity to control what happens in the case that pre-pruning and post-pruning subtrees have the same accuracy.