TDT4171 Methods in AI

Assignment 4

Written Spring 2022 by Ø. Solbø

Learning decision trees

Two methods for choosing the 'optimal' attribute are used during this assignment. The first method randomly selects an attribute to split on. Thus effectively giving the attributes random weights. The second method tries to choose the attribute which maximizes the expected information gain, as described in [1, p. 680]. Theoretically, one expects the random tree to behave worse than the tree being developed using the expected information gain, because selecting attributes randomly may not induce further information. Thus, the tree should be expected to become longer, as more attributes are necessary to filter the data. In other words, weighting the attributes randomly, causes the worst case behaviour of the system to be linear in complexity. Thus, the tree could be relatively inefficient for larger data sets. The tree using the expected information gain on the other hand, is expected to better separate the data, causing the tree to be shorter and more efficient to search through.

Somehow, the developed code does not behave correctly when plotting the tree. No explanation has been found to why only two of the leaf-nodes are plotted. Looking through the data in a debugger, the leaf-nodes exists, but are somehow not shown using graphviz. The trees themselves, therefore do not give enough visual information, as one cannot look directly at a tree and know the obtained result. If the algorithm was meant to be developed further, a proper visualization would be crucial for debugging, and for understanding the data better.

Some of the trees developed using the random weighting, are shown in figure 1. The trees do clearly vary between the corresponding runs, which causes them to have slightly different performance. The tree(s) developed for maximizing the expected information gain, is shown in figure 2. Independent of runs, only one tree is generated using the training data, due to consistency in the data and the implementation. A direct comparison, shows this tree to be much shorter compared to the randomly selected trees. This implies that each attribute is a better separation of the available information, thus not requiring the algorithm to potentially check every single attribute. The algorithm using the expected information gain to select the attributes, is therefore much more efficient and more likely to generalize better.

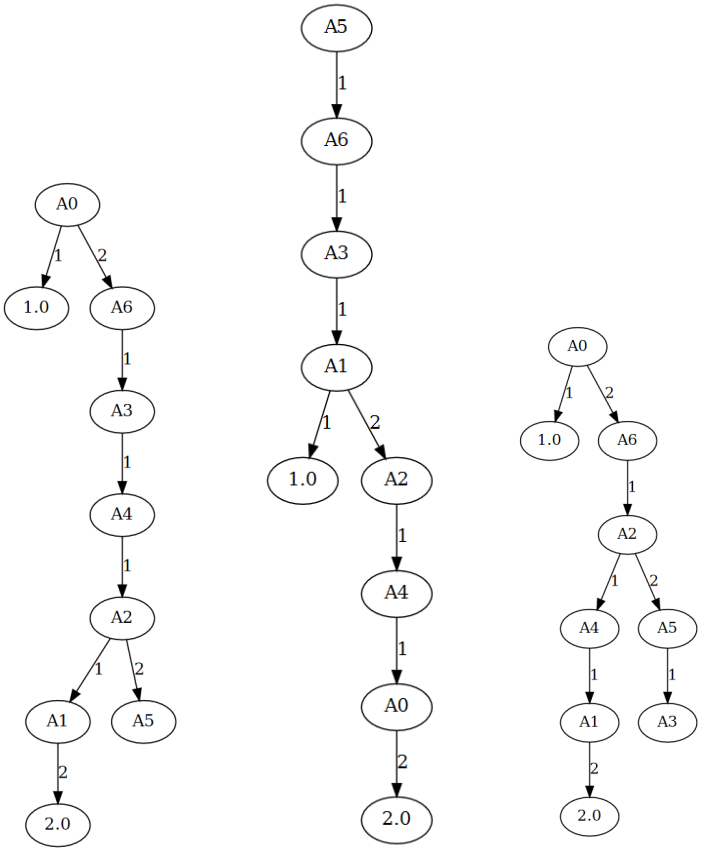

Figure 1: Three runs when using random weighting to select the attributes.

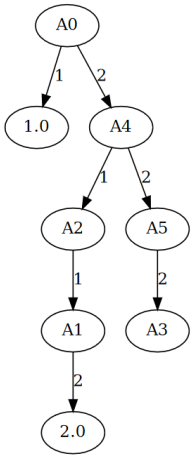

Figure 2: The developed tree using the expected information gain to select the attribute.

To compare the performance off the different methods, 25000 iterations were run. At each run, a new tree was generated from the training-data, and then compared to the test-data. A histogram showing number of correct for each method, is shown in figure 3, with the corresponding statistics is shown in table 1.

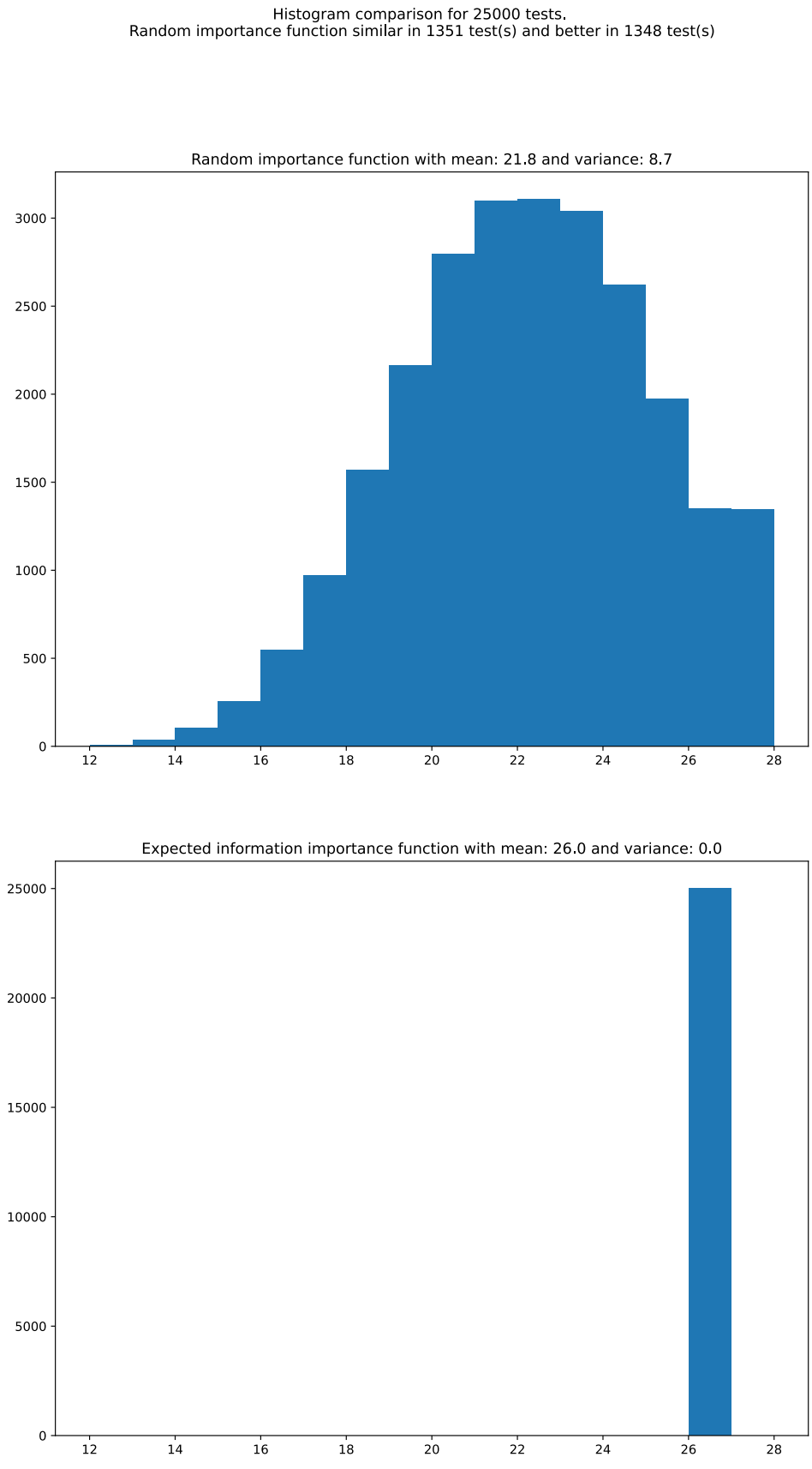

Figure 3: Histogram and some statistics for the different methods.

Method	Mean	Variance
Random	21.8	8.7
E[information gain]	26.0	0.0

Table 1: Mean and variance for correct tests for the two methods.

Out of 25000 tests, the random method behaved similar or better 1351 and 1348 times respectively. In all other tests, the decision tree choosing attributes based on the expected information gain behaved better. Although some of the random iterations gave more correct predictions, it did not occur consistently. Due to the sheer number of obtainable trees, having an algorithm that produces consistent and 'good enough' results, is far more preferable to an algorithm that sometimes an optimal result. This would become even more crucial for a more complex system.

To test the performance of the different codes, the number of CPU-cycles where also calculated, with the results shown in table 2. These numbers may vary between CPUs. The performance measurement includes both creation, documentation and testing of the different trees over 25000 tests. The tree using the expected information gain, is in general much more efficient. As explained earlier, the chosen attributes manages to split far more information, causing fewer attributes to be observed before reaching a conclusion. As observed in figure 1, the random trees may induce a worst-case time complexity which is linear in number of attributes. For a small problem like this, choosing randomly has little impact on the performance, but a larger problem could suffer immensely from this.

Method	Mean	Variance
Random	9717630.2	3205200450959.8
E[information gain]	8876840.7	73595193210.8

Table 2: Mean and variance for CPU-cycles for the two methods. Includes both creation, documentation and testing of the different trees over 25000 tests, obtained using an AMD Ryzen 7 5700x.

Based on the histogram, two questions arise. Firstly, why does the random attribute sometimes outperform the choice of attribute using the expected information gain? The expected result would be for the expected information gain to behave better. For the random method to function better in some cases, it means that there is some dynamics which the expected information gain cannot capture. This might be caused when multiple attributes would contribute equivalently in one iteration, but give slightly different trees in the subsequent iterations, however this behaviour has never been observed.

Secondly, why do some of the randomly selected attributes produce test-results below 50 %? The expected result was for the random method to give slightly worse results, but not be so terrible. In some cases, just guessing the type would be far more computational efficient and have better expected performance.

Although the reasons behind said observations are not understood, it may be caused by terrible code. During the development, some dubious decisions were made, which caused the entire project to become far too complicated. Too many bugs were created by python's use of references instead of const ref, and there might be some remaining impacting the results. One could therefore question the validity of the results, however no direct material supporting this conclusion could be directly pointed out.

References

[1] S. Norvig, P. Russel, *Artificial Intelligence - A modern approach.* Pearson, 2022.