KTH Royal Institute of Technology

Lab 1: Decision Trees

DD2421 Machine Learning

Yasmin Baba ybaba@kth.se
Héloïse Dehem dehem@kth.se

20th September 2019

Table of Contents

2 MONK DATASETS **3**

ASSIGNMENT 0: 3

3 ENTROPY **3**

ASSIGNMENT 1: 3
ASSIGNMENT 2: 3

4 INFORMATION GAIN **4**

ASSIGNMENT 3: 4
ASSIGNMENT 4: 4

5 BUILDING DECISION TREES **5**

ASSIGNMENT 5: 5

6 PRUNING **5**

ASSIGNMENT 6: 5
ASSIGNMENT 7: 6

2 MONK datasets

Assignment 0:

Each one of the datasets has properties which makes them hard to learn. Motivate which of the three problems is most difficult for a decision tree algorithm to learn.

MONK-2 is the most difficult for a decision tree algorithm to learn. We have to check all the attributes.

3 Entropy

Information gain is measured in terms of the expected reduction in the entropy (impurity) of the data, where entropy is:

$$Entropy(S) = -\sum_i p_i \log_2 p_i$$

p_i – proportion of examples of class i in S

The monk dataset is a binary classification problem and thus the equation simplifies to:

$$Entropy(S) = -p_0 \log_2 p_0 - p_1 \log_2 p_1$$

Where p_0 and $p_1 = 1 - p_0$ are the proportions of examples belonging to class 0 and 1

Assignment 1:

The file dtree.py defines a function entropy which calculates the entropy of a dataset. Import this file along with the monks datasets and use it to calculate the entropy of the training datasets.

```
Entropy for monk1 :  1.0
Entropy for monk2 :  0.957117428264771
Entropy for monk3 :  0.9998061328047111
```

MONK-1: equal probability of outcomes so entropy is highest

Assignment 2:

Explain entropy for a uniform distribution and a non-uniform distribution, present some example distributions with high and low entropy.

In a uniform distribution, outcomes have equal probability (like a fair coin, perfect die, …)

Examples:
if $p_0 = \frac{1}{2}$ and $p_1 = \frac{1}{2}$

$$Entropy(S) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = -\log_2\frac{1}{2} = \log_2 2 = 1$$

if $p_0 = \frac{1}{n}$, $p_1 = \frac{1}{n}$, … ,$p_n = \frac{1}{n}$

$$Entropy(S) = -\frac{1}{n}\log_2\frac{1}{n} - \frac{1}{n}\log_2\frac{1}{n} - \cdots - \frac{1}{n}\log_2\frac{1}{n} = -\log_2\frac{1}{n} = \log_2 n$$

Therefore for uniform distributions, the entropy grows logarithmically with the amount of classes. A fair coin toss, has thus a lower entropy than a perfect die, it is more predictable.

In a non-uniform distribution, outcomes have unequal probabilities (bias coin, bias die, …). There will be one outcome that will be more probable than the others and thus the outcome is less uncertain (more predictable) than the uniform case.

Therefore entropy of a non-uniform distribution will always be less than the uniform distribution of same dimension n:

$$Entropy(S) \leq \log_2 n$$

4 Information Gain

The information gain measures the expected reduction in impurity caused by partitioning the examples according to an attribute. It thereby indicates the effectiveness of an attribute in classifying the training data.

$$Gain(S, A) = Entropy(S) - \sum_{k \,\in\, values(A)} \frac{|S_k|}{|S|} Entropy(S_k)$$

S_k – the subset of examples in S for the attribute A has the value k

Assignment 3:

Use the function averageGain (defined in dtree.py) to calculate the expected information gain corresponding to each of the six attributes. Note that the attributes are represented as instances of the class Attribute (defined in monkdata.py) which you can access via m.attributes[0], ..., m.attributes[5]. Based on the results, which attribute should be used for splitting the examples at the root node?

Dataset	a_1	a_2	a_3	a_4	a_5	a_6
MONK-1	0.07527	0.00584	0.00471	0.02631	0.28703	0.00076
MONK-2	0.00376	0.00246	0.00106	0.01566	0.01728	0.00625
MONK-3	0.00712	0.29374	0.00083	0.00289	0.25591	0.00708

Assignment 4:

For splitting we choose the attribute that maximizes the information gain, Eq.3. Looking at Eq.3 how does the entropy of the subsets, Sk, look like when the information gain is maximized? How can we motivate using the information gain as a heuristic for picking an attribute for splitting? Think about reduction in entropy after the split and what the entropy implies.

To maximise $Gain(S, A)$ the entropy of the subsets S_k must be as small as possible. This makes sense as the uncertainty in the subset will decrease the more information we have.

So we should choose an attribute with the highest information gain to split, resulting in subsets with the lowest uncertainties. These subsets will therefore have lower entropy – we can therefore move towards a limited number of subsets that contain better classified samples.

5 Building Decision Trees

Assignment 5:
Build the full decision trees for all three Monk datasets using buildTree. Then, use the function check to measure the performance of the decision tree on both the training and test datasets.
Compute the train and test set errors for the three Monk datasets for the full trees. Were your assumptions about the datasets correct? Explain the results you get for the training and test datasets.

	$\boldsymbol{E_{train}}$	$\boldsymbol{E_{test}}$
MONK-1	0	0.17129629629
MONK-2	0	0.30787037037
MONK-3	0	0.05555555555

MONK1 and MONK2 have errors higher than expected.

The training error is zero because we are building trees that classify perfectly, producing good results on the training data, but the model generalizes poorly.

Our assumptions about the dataset monk-2 were right, it is the most difficult for a decision tree algorithm to learn, because all attributes have to be checked to have an exact model.

6 Pruning

The idea of reduced error pruning is to consider each node in the tree as a candidate for removal. A node is removed if the resulting pruned tree per- forms at least as well as the original tree over a separate validation dataset, i.e. a dataset not used during training. When a node is removed, the sub- tree rooted at that node is replaced by a leaf node, to which the majority classification of examples in that node is assigned.

Assignment 6:
Explain pruning from a bias variance trade-off perspective.

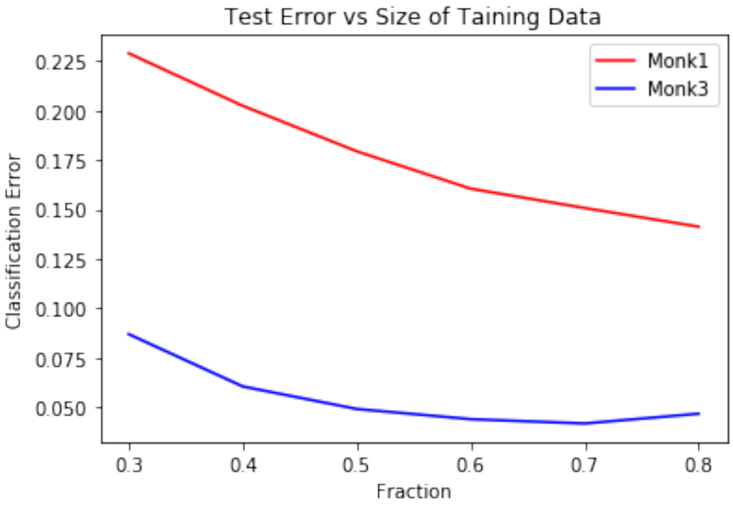
Decision trees tend to be complex models, having low bias and high variance. It has low bias because the tree is highly tuned to the data present in the training set. It has a high variance as when a new data point is fed, even if one of the parameters deviates slighting, the condition will not be met and it will take the wrong branch. The deeper the tree, the lower the bias and the higher the variance.

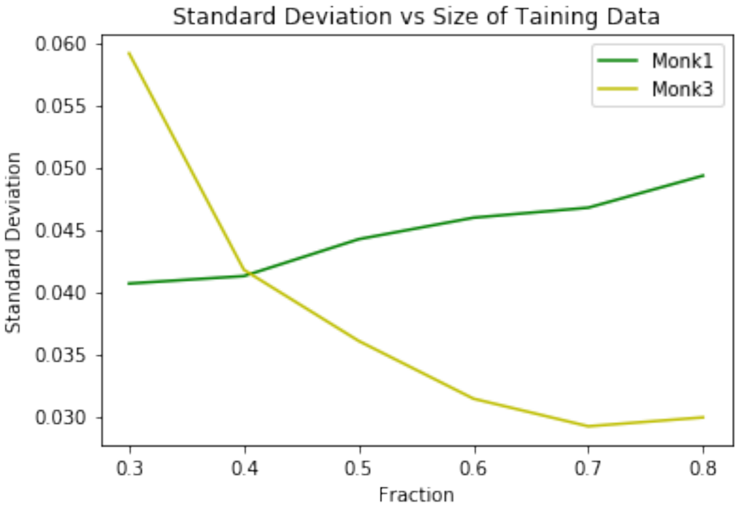
By pruning the tree and removing overfitting nodes/leaves, the bias increases (simplifying assumptions made) and the variance reduces.

Assignment 7:

Evaluate the effect pruning has on the test error for the monk1 and monk3 datasets, in particular determine the optimal partition into training and pruning by optimizing the parameter fraction. Plot the classification error on the test sets as a function of the parameter fraction 2 {0.3, 0.4, 0.5, 0.6, 0.7, 0.8}.

Note that the split of the data is random. We therefore need to compute the statistics over several runs of the split to be able to draw any conclusions. Reasonable statistics includes mean and a measure of the spread. Do remember to print axes labels, legends and data points as you will not pass without them.


Test Error vs Size of Taining Data
Classification Error
Fraction
Monk1
Monk3


Standard Deviation vs Size of Taining Data
Standard Deviation
Fraction
Monk1
Monk3