Lab 1 - Machine Learning

Irene Natale

July 27, 2018

Datasets presentation

the problem involves three different binary classification problems, MONK-1, MONK-2, MONK-3, each of them includes six discrete attributes. The attributes can take the following values

$$a_1 \in \{1,2,3\} \quad a_2 \in \{1,2,3\} \quad a_3 \in \{1,2\}$$

$$a_4 \in \{1,2,3\} \quad a_5 \in \{1,2,3,4\} \quad a_6 \in \{1,2\}$$

Further characteristics of the datasets (training and test sets) can be found in Table 1 and 2 of the assignment.

Assignment 0

The first thing to notice is that the training datasets are small compared to the test datasets, and this is not positive. This makes the three data sets generally hard to learn. Furthermore, all of the three monk datasets have characteristics that are hard to be represented with a decision tree. In a decision three we usually take an attribute and a threshold and we divide the dataset into the element that have the attribute value below the threshold and the elements that have it above. This procedure does not represents any of the three MONK datasets we are considering.

Assignment 1

We use the function "entropy" to compute the entropy of each training dataset. Results are shown in Table 1.

Assignment 2

A uniform distribution is a distribution is which every outcome has the same probability to happen. Classic examples can be a fair dice, and the actual

Table 1:

Dataset	Entropy
MONK-1	1.0
MONK-2	0.957117428265
MONK-3	0.999806132805

Table 2: Information Gain

Dataset	a1	a2	a3	a4	a5	a6
MONK-1	0.0752725	0.0058384	0.0047075	0.0263116	0.2870307	0.0007578
MONK-2	0.0037561	0.0024584	0.0010561	0.0156642	0.0172771	0.0062476
MONK-3	0.0071208	0.2937361	0.0008311	0.0028918	0.2559117	0.0070770

continuous uniform distribution. The latter has the following probability density function

$$f(x) = \begin{cases} \frac{1}{b-a}, & a < x < b \\ 0, & \text{otherwise} \end{cases}$$

and the entropy for this case will be computed as follows

$$E = -\int_a^b \frac{1}{b-a} log(\frac{1}{b-a}) dx = \int_a^b \frac{1}{b-a} log(b-a) dx = log(b-a)$$

When the domain largeness becomes wider, the entropy gets higher as well. This makes sense, since the entropy measures the total variance across classes. In a non-uniform distribution, in which one or more outputs are more likely to happen then the others, the entropy behaviour is different. If we take for example the Bernoulli distribution, which has the following probability density function

$$f(x) = \begin{cases} p, & x = 1 \\ 1-p, & x = 0 \\ 0, & \text{otherwise} \end{cases}$$

the entropy computation becomes

$$E = -plog(p) - (1-p)log(1-p)$$

The entropy in this case takes its maximum value 1 when $p = 0.5$. In fact, in this case we have the same probability of getting value 0 or 1, and consequently a high variance of values. If, for example, we take $p = 0.2$, the output $x = 0$ is definitely more likely to happen then $x = 1$. This means that the output $x = 0$ contains a smaller amount of information, while the output $x = 1$, which is less likely to happen, contains a higher amount of information. In this case, if we consider several outputs, we will obviously obtain more $x = 0$ then $x = 1$, and so the variance of output values will be smaller then the previous case with $p = 0.5$.

Assignment 3

Look at Table 2. Since the Information Gain indicates the effectiveness of an attribute in classifying the training data, we need to select the argument that leads to the higher Information Gain. Therefore, the fifth argument is chosen for MONK-1 and MONK.2, while the second argument is selected for MONK-3.

Table 3: Fraction of correctly classified samples

	Train samples	Test samples
MONK-1	1.0	0.8287037
MONK-2	1.0	0.6921296
MONK-3	1.0	0.9444444

Table 4: My caption

	Train samples error	Test samples error
MONK-1	0.0	0.1712962
MONK-2	0.0	0.3078703
MONK-3	0.0	0.0555555

Assignment 4

We take into consideration equation (3):

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

It is clear that when the entropy of the subset S_k decreases, the information gain increases. The maximum value of information gain corresponds to the minimum value of entropy S_k. If the considered split "works well", it will lead to a more homogeneous subset of values, with a lower entropy. Another way of seeing this is that if the chosen split is effective we are using a big amount of information (measured my the information gain).

Assignment 5

With

$$tree1 = d.buildTree(m.monk1, m.attributes)$$

$$tree2 = d.buildTree(m.monk2, m.attributes)$$

$$tree3 = d.buildTree(m.monk3, m.attributes)$$

the function "buildTree" is used to create the decision tree for the three training datasets MONK-1, MONK-2 and MONK-3. It is possible to compute the fraction of correctly classified samples for every tree. Table 3 shows the obtained results. Table 4 instead shows the values for the errors.

The first thing that pops out from the results is that the errors coming from the train datasets are all 0.0. This means that the generated tree is perfectly fitting the train datasets. However the errors do not stay low for all the test datasets. In fact, for MONK-1 and MONK-2 are pretty high ($\approx$ 0.17 and $\approx$ 0.31 respectively). This is a clear case of overfitting, where the found scheme fits really well the train data sets, but not the test data sets. This is due to a model that is not able to capture the real structure of data because it is overly specialized for the training samples. In conclusion, the assumptions about the datasets are incorrect, and we are not able to represent the datasets concepts with the decision trees that we constructed.

Table 5: Error mean and variance for MONK-1

training fraction	0.3	0.4	0.5	0.6	0.7	0.8
monk1 mean	0.234701	0.206053	0.175112	0.156120	0.142026	0.125960
monk1 variance	0.054068	0.059863	0.062535	0.064660	0.069641	0.073873

Table 6: Error mean and variance for MONK-3

training fraction	0.3	0.4	0.5	0.6	0.7	0.8
monk3 mean	0.132465	0.109324	0.095868	0.088367	0.079567	0.070760
monk3 variance	0.054846	0.043255	0.035193	0.033341	0.040131	0.046538

Assignment 6

In this section I created a code that calls the function "allPruned" and picks the pruned tree that gives the best classification performance on the validation test. In this case the full tree is created from the train test and then the classification performance is computed on the validation test. The function "partition" divides the original training data samples into training and validation sets.
It is possible to analyze the pruning process from a bias variance trade-off perspective. In the case we are considering, the whole decision tree (before the pruning process) represents the most complex model we can consider, because of the highest number of leaves. When we start pruning the tree, we choose to use a less and less complex model. Ideally, the model we aim for needs to have low variance and low bias. If we choose a model that is too complex, we will obtain high variance. Instead, if we choose a model that is too simple for the phenomena we are considering, we will obtain a model with a high bias. Therefore, the tricky aspect of what we are doing is to find the point of balance between these two aspects. This can be done using a validation dataset, generated from the original training dataset.

Assignment 7

In order to understand which is the best division of data into training and test subsets we need to compute some calculations. For the first and third monk datasets we divide the whole training datasets into training and validation subsets with different proportions ($[0.3, 0.4, 0.5, 0.6, 0.7, 0.8]$) and we look at the generated error. The split of data is performed by the function "partition" and it is done randomly. For this reason we run the same computations 1000 times and look at the mean and spread of values. The mean is computed as

$$mean = \frac{1}{N} \sum_{i=0}^{i=N} v_i$$

and the variance of data as the standard deviation. Results are shown in Table 5 and 6 and in Figures 1 and 2

For both cases the error mean is lower when the training set is composed of 80 percent of the whole data (fraction = 0.8).

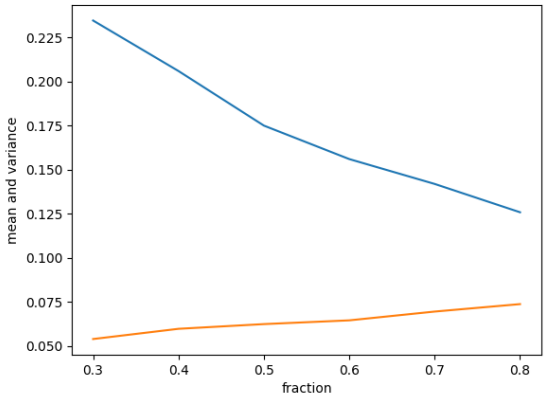

Figure 1: Error mean (in blue) and error variance (in orange), computed on the validation dataset for the monk-1 case

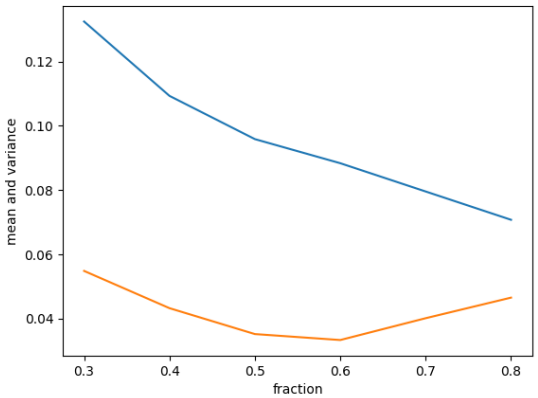

Figure 2: Error mean (in blue) and error variance (in orange), computed on the validation dataset for the monk-3 case