Machine Learning

Exercise Sheet 9

Winter Term 2023
Prof. Dr. Niels Landwehr
Dr. Ujjwal

Available: 19.01.2024
Hand in until: 26.01.2024 11:59am
Exercise sessions: 29.01.2024/31.01.2023

Task 1 – Information gain for continuous random variable [10 points]

Let us consider a continuous-valued variable $A \in \mathbb{R}$ following a normal distribution with mean μ and variance σ^2, that is, $A \sim \mathcal{N}(\mu, \sigma^2)$. For a continuous variable, we can define the entropy by replacing the sum in the definition of the discrete entropy by an integral:

$$H(x) = -\int p(x)\ \ln\ p(x)dx$$

a) Compute the entropy of the random variable A (as a function of σ and μ).
b) Determine the values of σ for which the entropy calculated above becomes negative.

Task 2 – Decision tree from partition [5 points]

Figure 1 shows a partition of the space $\mathbb{R}^2$ into cells. Using splits of the form $x_i \leq C$, construct a decision tree on the attributes x_1, x_2 such that the cells in the partition correspond to the leaves in the tree. You do not need to use any learning algorithm, just construct the tree manually. Use the convention that the left branch is followed if the split condition is satisfied and the right branch otherwise.

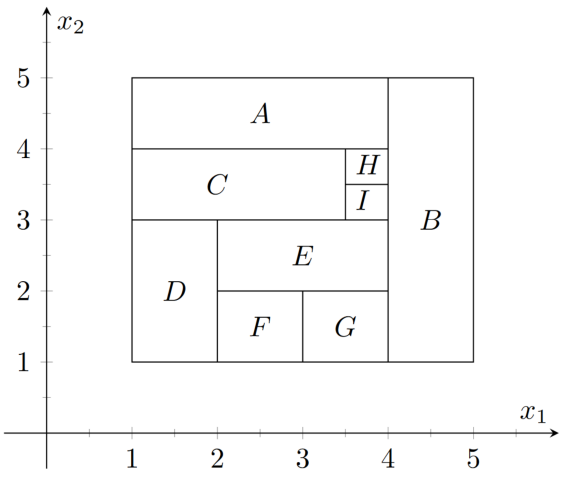

Figure 1: A partition of the space $\mathbb{R}$.

Task 3 – Decision tree learning with information gain [10 points]

Consider the small toy data set about the relationship between species (Asian elephant *Elephas maximus* and the African elephant *Loxodonta africana*), weight and gender of elephants given by the following table:

Weight (x_1)	Species (x_2)	Gender (y)
5500kg	African	male
3500kg	African	female
3400kg	Asian	male
2700kg	Asian	female

From this data, we want to learn a model that predicts the gender of an elephant from its weight and species.
Manually learn a decision tree from this data, using splits of the form $x_1 < C$ for the attribute x_1 and binary splits for the nominal attribute x_2. Use the algorithm from Slide 30 of the decision tree lecture, with $N_{min} = 1$ and the Gini-index for selecting the splits. Draw the learned tree. Use the convention that the left branch is followed if the split condition is satisfied and the right branch otherwise.
By hand, draw a minimal-depth tree that also solves the problem with zero error on the training data. Why is this shallower tree not found by the learning algorithm?

Task 4 – Programming Decision Trees [15 points]

You are provided an IPython notebook *trees.ipynb*. A number of basic functions to implement decision trees are already implemented in the notebook. The main function is *dtree* which implements a basic decision tree algorithm. Your tasks are the following:

1. Complete the function *determine_best_split*() which selects the column and the split value to do the splitting. Use entropy as the metric for implementation for which the functions are already implementned.

2. Run the algorithm for two datasets – a) banknote authentication dataset and b) iris dataset. The files for these datasets are already provided to you. Please follow the instructions in the notebook and evaluate the algorithm for these datasets using a 70%-30% train-test split.

3. Use DecisionTreeClassifier in scikit-learn to learn trees for these two data sets and compare them to the output of the algorithm in the notebook. Comment on the differences and optimality.