Data Mining Summer Term 2020

Institute of Neural Information Processing
PD Dr. F. Schwenker

Assignment 10 (Submission until July 14, 2020)

Exercise 1 (3 points): Impurity Measures [Pen and Paper]

An impurity measure $Q(R)$ of a set of labelled examples R estimates how clean the set is with respect to the number of different class labels in R. We get high impurity values if R contains data points from a lot of different classes and low impurity values if nearly all points belong to the same class. Here, we take a look at some concrete measures and in the next exercise, we use them to build a decision tree.

In order to calculate the impurity of a set R, we first need to calculate the corresponding probability vector $\boldsymbol{p} = (p_1, p_2, \dots, p_L)$ with the relative frequencies $p_i = n_i/|R|$. They are based on the number of examples n_i in the set R which belong to the class i. There are L classes in total. We know the following three impurity measures from the lecture:

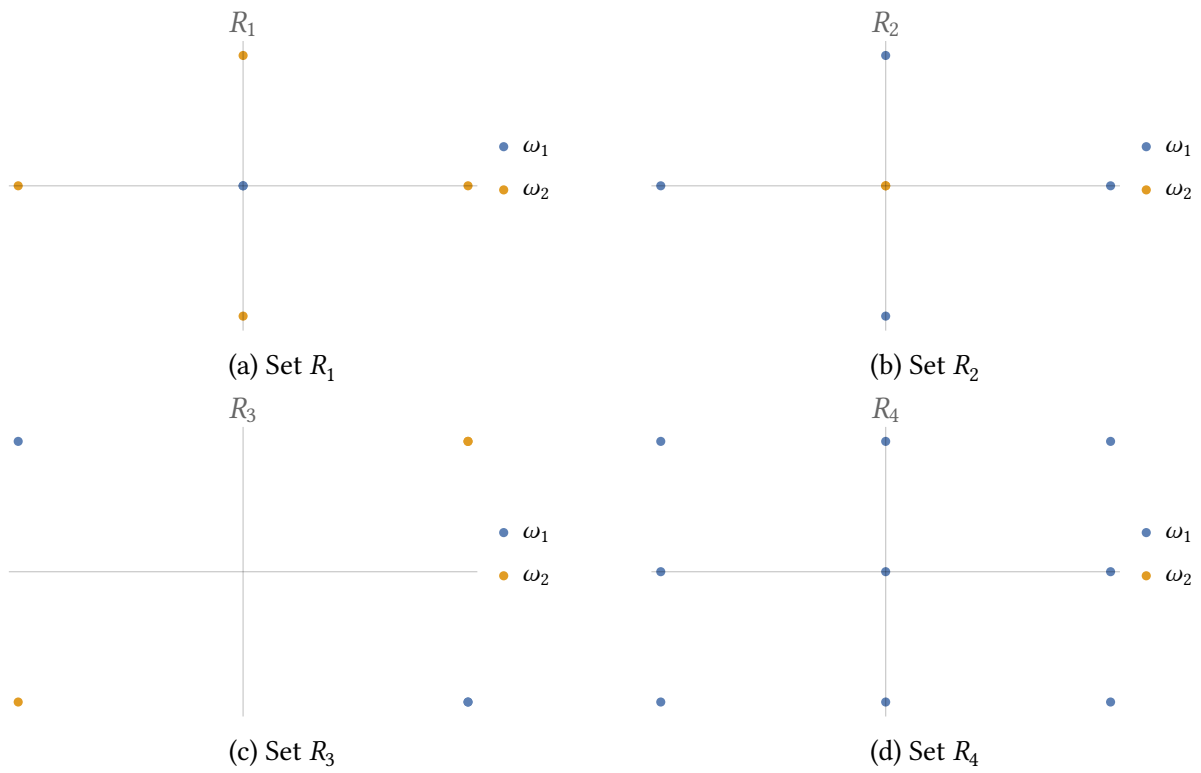
$$Q_m(\boldsymbol{p}) = 1 - \max_{i=1}^{L}(p_i) \qquad \text{misclassification index} \tag{1}$$

$$Q_g(\boldsymbol{p}) = 1 - \sum_{i=1}^{L} p_i^2 \qquad \text{Gini index} \tag{2}$$

$$Q_e(\boldsymbol{p}) = -\sum_{i=1}^{L} p_i \cdot \log_2(p_i) \qquad \text{entropy index} \tag{3}$$

All three measures fulfil Breiman's conditions (script page 226).

1. In the special case of $L = 2$ we can write the probability vector as $\boldsymbol{p} = (p_1, 1 - p_1)$ so that the impurity measures reduce to a function of the variable p_1. Figure 2 shows a plot for each impurity measure for this case. Map each line of Figure 2 (A, B or C) to one of the impurity measures (Equation 1, Equation 2 or Equation 3).
2. In Figure 1, four different sets with points from two classes ($L = 2$) are shown. You can think of each point as a labelled example. For the impurity measures, we are interested in the relation of the number of points from one class compared to the total number of points. For each set R_i, fill out the corresponding row of Table 1, i.e.
   a) name the probability vector $\boldsymbol{p}_i$ and
   b) calculate the value for all three impurity measures. If you use a shortcut in the calculations, name the property of the impurity measure you took advantage of.



(a) Set R_1 (b) Set R_2

(c) Set R_3 (d) Set R_4

Figure 1: Four different sets R_i with points from two classes. None of the points overlap with each other.

Table 1: Probability vector $\boldsymbol{p}_i$ and impurity measures for the sets shown in Figure 1.

Set R_i	$\boldsymbol{p}_i$	$Q_m(\boldsymbol{p}_i)$	$Q_g(\boldsymbol{p}_i)$	$Q_e(\boldsymbol{p}_i)$
R_1				
R_2				
R_3				
R_4				

Exercise 2 (7 points): Decision Tree [Pen and Paper]

Decision trees are popular classifiers widely used in supervised learning. They basically derive a class label by asking simple questions concerning one feature at a time. Among the main advantages is that they can handle arbitrary scaled input data and have a representation which is easily readable for humans. In this exercise, we want to take a look at how we can construct such a decision tree based on the known impurity measures.

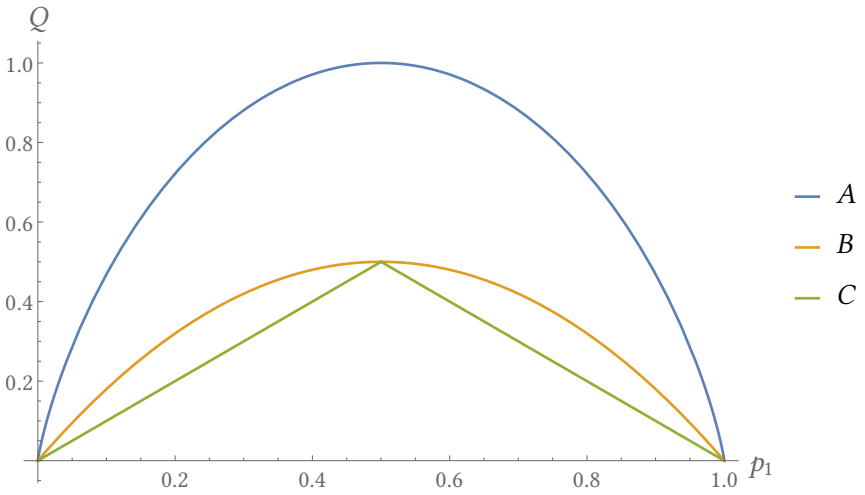


Figure 2: Plot of the impurity measures as a function of p_1.

The basic procedure is to split the tree repeatedly until all data points are classified correctly. In each step, we test every feature and decide about how good the split would be by calculating the gain of the split. We then perform the split on the feature with the highest gain. This process is repeated recursively. Based on the impurity measures, we can define the impurity gain as

$$\Delta Q(R, R_1, R_2, \dots, R_B) = Q(R) - \sum_{i=1}^{B} p_{R_i} Q(R_i). \tag{4}$$

R is the set of data points in the root node and R_i the remaining sets when splitting on the feature value i (based on a feature with B possible values). The impurity for each feature value is weighted by the relative frequency $p_{R_i} = |R_i|/|R|$. This attaches higher importance to feature values which cover more data points. As concrete impurity measure, we want to use the misclassification index (Equation 1) here.

Our goal is to find a decision tree for the dataset shown in Table 2. It classifies parties as either *Hit* or *Flop* based on three features measured over seven observations $\boldsymbol{x}_i$. The first feature is the measured temperature converted to the feature values *Cold* or *Warm*. The second feature measures the number of guests which attended the party and the third feature tells us something about the major served food.

1. Regarding the level of measurement of the features of Table 2: what scale do they have?
2. Let's begin and construct our decision tree. We need to check each feature and measure the impurity gain. Then, we split by the feature leading to the highest gain.
    a) Before we begin analysing the features, we need some information about the root set R. Since we have not performed any splits so far, this set contains all data points, i.e. $R = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_7\}$. Calculate the probability vector $\boldsymbol{p}$ and the impurity $Q_m(R)$ for this set.

b) Temperature is the first feature and it is a binary feature.

   i. Set up the two sets R_1 and R_2.

   ii. Calculate the relative frequency p_{R_i} for each set.

   iii. Calculate the probability vectors $\boldsymbol{p}_i$ and the impurity measures $Q_m(R_i)$.

   iv. Now, we have all ingredients together so that you can calculate the impurity gain as defined by Equation 4 for this split.

c) The next feature is the number of guests and this one is a bit trickier since the feature values are not from a finite set. Rather, every non-negative integer number is possible. What we want is to find a threshold θ so that we retrieve one set R_1 with $g < \theta$ and another set R_2 with $g \geq \theta$ (g denotes the number of guests).

   i. There are several approaches to find the threshold θ. Here, we use a simple "mean of the means" procedure. Regarding the number of guests feature, calculate the mean μ_1 for all data points from the *Hit* and the mean μ_2 for all points from the *Flop* class. Then, calculate the midpoint between these two values, i.e. the mean $\theta = 0.5 \cdot (\mu_1 + \mu_2)$. Apply this threshold and retrieve the two sets R_1 and R_2.

   ii. Proceed with the remaining steps as before to calculate the impurity gain for the second feature.

d) The third feature, served food, is similar to the temperature feature only that we now have three sets R_1, R_2 and R_3. Calculate the impurity gain for this feature.

e) Decide about which feature to use, i.e. which split to perform.

3. Repeat the previous procedure recursively for each new set R_i obtained from the split. That is, each R_i becomes the root element and then check the relevant features again by calculating the impurity gain and deciding about the next split. Continue until all leaf nodes contain only data points from one class. In the end, every data point should be classified uniquely by the decision tree.

4. Draw the final decision tree. At each node, name which feature to check and on the edges write the feature values (or threshold boundaries). The leaf nodes should contain the classified data points $\boldsymbol{x}_i$.

Table 2: Example dataset where we want to build a decision tree from.

Data point	Temperature	Number of guests	Food	Class
x_1	Cold	10	Nothing	Flop
x_2	Cold	20	Vegetables	Hit
x_3	Cold	2	Vegetables	Flop
x_4	Cold	8	Snacks	Hit
x_5	Warm	30	Snacks	Hit
x_6	Warm	5	Nothing	Flop
x_7	Warm	28	Nothing	Hit