MACHINE LEARNING 1 WS2019/20
3. EXERCISE

- Classification -

problem 1. Decision Trees (1)

Consider a data set comprising 400 data points from class C1 and 400 data points from class C2. Suppose that a tree model A splits these into (300, 100) at the first (i.e. C1-) leaf node and (100, 300) at the second (C2-) leaf node, where (n,m) denotes that n points are assigned to C1 and m points are assigned to C2 in the training set. Similarly, suppose that a second tree model B splits them into (200, 400) and (200, 0).

Evaluate the misclassification rates for the two trees and hence show that they are equal. Similarly, evaluate the Information gain / Gini impurity / impurity pseudo metric / (Gini coefficient) for the two trees and compare it.

Entropy:

$$H(T) = \sum_{i=1}^{c} -p_i log_2(p_i) \tag{1}$$

- p_i : fraction of examples belonging to class i
- c : number of classes (e.g., 3 in the fruits example)
- Define $0 \cdot log(0) = 0$ (if no more examples of a class left)

Information gain (for n-ary attribute A, we consider n=2 = binary attributs):

$$IG(A) = H(T) - \sum_{a=1}^{n} \frac{|T_a|}{|T|} H(T_a) \tag{2}$$

Gini impurity:

$$I_G(T) = \sum_{i=1}^{c} p_i(1 - p_i) = \sum_{i=1}^{c} (p_i - p_i^2) = 1 - \sum_{i=1}^{c} p_i^2 \tag{3}$$

Gain is again calculated as the weighted average (as for IG).

Impurity pseudo metric (no weighted average):

$$\Delta(y, y') = \frac{2 \cdot |G(y, y')|}{|S|(|S| - 1)} \tag{4}$$

After the split, there are four possibilities:
1. All remaining samples belong to one class. Label the leaf with the class.
2. The samples belong to a mixture of classes. Continue to split the samples recursively with a new best attribute
3. If there are no attributes left, label the node with the majority classification for the remaining samples
4. If there are no samples left, label the node with the majority classification of the parent node

problem 2. Decision Trees (2)

Run a Decision Tree example from Scikit-learn.
Use that example to build a DT for datasets Data3.csv and Data4.csv.

Develop your own Decision Tree code for discrete attributes. Try it on e.g. the Covertype data.

We will discuss some aspects of Decision Trees based on US income data.
Here is code which uses only discrete labels. In addition it uses deeper to measure the memory consumption of the tree and the US income data.
Change one character of that code to improve its performance and size!
Explain why this works.
Change that code to work with the Covertype data.

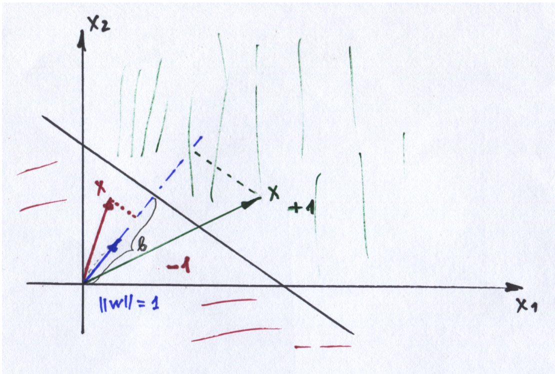
You may want to have a look at other DT codes

problem 3. Perceptron

Use the perceptron algorithm to determine a classifier for the dataset Data3.csv.

Perceptron - Problem:
(description by Dmitrij Schlesinger)

Building block for almost everything: a mapping $f : \mathbb{R}^n \to \{+1, -1\}$ – partitioning of the input space into two half-spaces that correspond to two classes



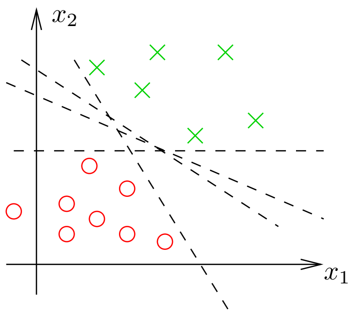
$$y = f(x) = \text{sgn}(\langle x, w\rangle - b)$$

with weights $w \in \mathbb{R}^n$ and a threshold $b \in \mathbb{R}$. Geometry: w is the normal of a hyperplane (given by $\langle x, w\rangle = b$) that separates the data. If $||w|| = 1$, the threshold b is the distance to the origin.

Let a training set $L = \big((x^l, y^l)\ldots\big)$ be given with
(i) data $x^l \in \mathbb{R}^n$ and
(ii) classes $y^l \in \{-1, +1\}$

Find a hyperplane that separates data correctly, i.e.

$$y^l \cdot [\langle w, x^l\rangle + b] > 0 \quad \forall l$$

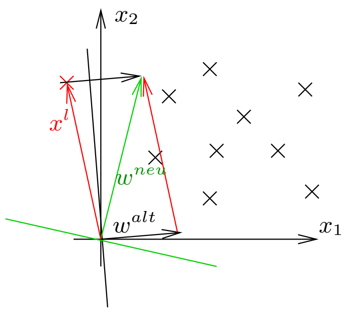


The task can be reduced to a system of linear inequalities:

$$\langle w, x^l \rangle > 0 \quad \forall l$$

Perceptron - Algorithm (Rosenblatt, 1958):

1) Search for an inequality that is not satisfied, i.e. an l so that $\langle x^l, w \rangle \leq 0$ holds;
2) If not found – End,
   otherwise, update $w^{new} = w^{old} + x^l$, go to 1).



The algorithm terminates after a finite number of steps (!!!), if there exists a solution. Otherwise, it never finishes.

problem 4. Logistic Regression

Use the Logistic Regression to determine a classifier for the datasets Data3.csv and Data4.csv.

problem 5. Neural Net

Use the simplest possible Neural Net to determine a classifier for the datasets Data3.csv and Data4.csv. Use Keras/Tensorflow or any similar environment.