Decision Tree Exercises - Part 1

Summary

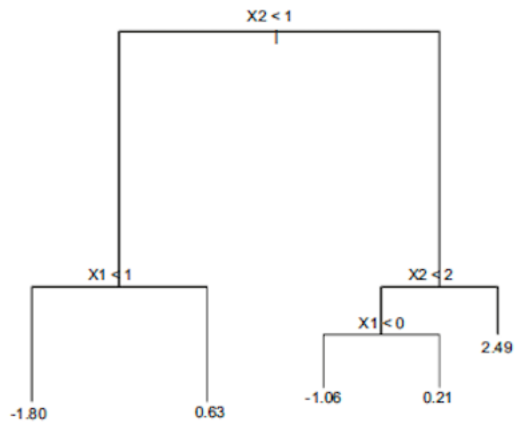
These exercises focus on understanding how a decision tree is built and interpreted.

Exercises

Exercise 1

- Sketch an example of your own invention of a partition of two dimensional feature space that could result from recursive binary splitting. Your example should contain at least six regions.
- Draw a decision tree corresponding to this partition. Be sure to label all aspects of your figures, including the regions, the cutpoints, etc.
- Can you express your decision tree as a set of nested rules for classification: If .. (if.. else..) .. else .. (if .. else ..) ..

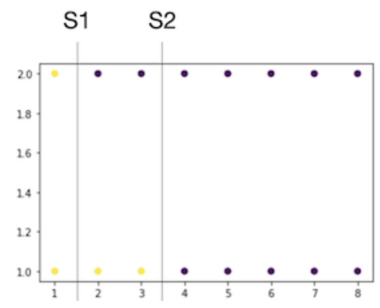
Exercise 2 Consider the following regression tree:



Sketch a partition [0,2]x[0,3] of a two dimensional feature space (X1 and X2) and divide it into the regions corrsponding to the tree, and indicate the mean for each region.

Exercise 3 Explain in your own words: What is a Decision Tree? How does it work? What are some similarities or differences to other algorithms you have used in this course?

Exercise 4 Consider the following simple classification setting with two classes, and S1 and S2 as two out of many possible splits of the data set.



Here we want to try three different impurity functions: classfication error rate, Gini index, and entropy.

Use all the impurity functions and compute by hand the **weighted average of impurities** of the child nodes, if we split the data by S1 and if we split it by S2.

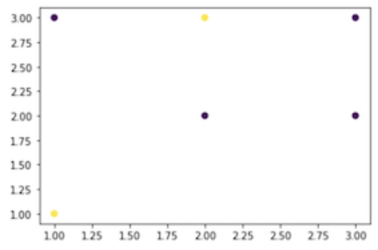
- Are the results differet for S1 and S2 when we use:
- classfication error rate?
- Gini index?
- Entropy?

One of the conditions we can use as stopping condition is to stop when further splitting does not reduce impurity (i.e., there is no information gain).

Use the above impurity functions on the whole dataset (before splitting).

- Based on each impurity function, is there any impurity reduction due to splitting the data by S1 or S2?
- Which of the impurity functions do you prefer to use? why?

Exercise 5 Consider the Gini index or entropy, and classfication error, in the following simple classification setting with two classes.

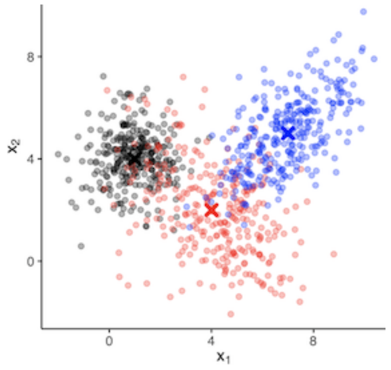


- Sketch a decision tree by hand for classifying this data.
- Now compute the impurity of the whole data (based on one of the impurity functions above) - you can do it by hand, but might be easier to automate

it by few lines of code :)

- Try every possible split, compute the impurity of each region and the weighted average of the children (or the information gain) for the split. What is the best split? Is it the same as the first split in your hand-made decision tree?
- Continue splitting based on the information gain until there is no classification error.

Exercise 6 Now use the good old datasets Ex1-training.csv and Ex1-test.csv :)



- Use DecisionTreeClassifier from sklearn to build a decision tree with maximum depth of 3.
- Visualize the tree and the decision boundaries. If you want, you may reuse/expand some code from Exercise_DT_setup.

Answer the following questions:

- Interprete the information shown in each node of the tree (threshold, gini, samples, value)?
- What is the accuracy score on the training set?
- How does the tree structure correspond to the decision boundaries?
- How does the decision boundary compare to what you have obtained from other classification methods in previous weeks e.g knn, LR, LDA and QDA? Why?

Exercise 7 Limiting the max_depth is a way of regularizing a decision tree, i.e. limit its complexity. The default setting (max_depth=None) allows the tree to grow until all leaves are pure or until another stopping criterion is reached.

- Re-run the code above with no limit on max_depth. What happens? How much of a difference does it make in this case?
- How deep must the tree be to reach 95% accuracy on the training data? (Note: this is just to understand how the decision tree works. For real

performance measures we would of course always use a test set).

- How well would you expect these trees to generalize?
- Plot training and test error for different max_depth. Does it match your expectation?

min_samples_leaf and max_leaf_nodes are among other parameters that can be used to regularize a decision tree.

- Find and read the documentation for these parameters, what do they do?
- Try varying their values on some of the trees trained above, does it affect these trees? How? Why?

Exercise 8 In this exercise, we train a regression tree for a 1-D feature space.

```
def regr1D(N):
    """ Returns 1D regression (x,y) dataset """

    global seed

    x = np.random.uniform(low= -50, high=50, size=(N,))
    y = 0.02 * x ** 3 + x ** 2 - 70 * x - 3
    y = y + (np.random.randn(N) * 50) #add noise

    return x,y

np.random.seed(42)
X,y = regr1D(100)
X = X.reshape(-1, 1)
```

- Run the code above to generate a regression dataset
- Then train a regression tree using DecisionTreeRegressor() with max_depth=2 and visualize it.
- How do you interprete the tree?
- Train more regression trees for the same dataset, increasing the max_depth to 3 or higher. What happens?
- Do you see any obvious advantages or disadvantages in using a decision tree for regression tasks?