4. Decision Tree for Regression

Decision tress can not only be used for regression, but also used for regression. However, there are some difference as the dependent variable are no longer only 0 and 1:

- While selection a variable to spilt, the criterion is no longer something like the misclassification rate, but something measures the degree of fit for continuous variables like mean square error.
- At each leaf, the mean value of that sample is taken as the estimate / forecast instead of the majority vote like that case in classification, as this minimize the mean square error. (#1)

A regression decision tree is run (using Python, #2) with the UCI Boston Housing dataset, and the R-square is found to be 96.3%, which is quite high.

#1

The reason why the using the mean value is same as minimizing the mean square error is because:

$X^* = \text{argmin}_{X^*} \sum (X_i - X^*)^2 = \text{argmin}_{X^*} \sum (X_i^2 - 2X^* X_i + X^{*2})$; $df/dx^* = \sum (0 - 2X_i + 2X^*) = 0$

$\Rightarrow 2\sum X^* - 2\sum X_i = 2nX^* - 2\sum X_i = 0$; $X^* = \sum X_i / n$ = mean of X.

#2

```
import numpy as np
import scipy as sp
from sklearn import tree
data = np.loadtxt(open("housing_data.csv", "rb"), delimiter=";", skiprows=1)
data_X = data[: , 0:-1]
data_Y = data[: , -1]
DT1 = tree.DecisionTreeRegressor(criterion='mse', min_samples_leaf = 0.005)
DT1 = DT1.fit(data_X, data_Y)
Y_head = DT1.predict(data_X)
SS_total = sum((data_Y - np.mean(data_Y)) ** 2)
SS_res = sum((data_Y - Y_head) ** 2)
R_sq = 1 - SS_res / SS_total
print R_sq
```

5. Lazy version of eager Decision Tree

The eager decision tree learning algorithm ID3 is logically well defined and intuitive, but sometimes it is a bit slow. Therefore, some people suggested a lazy version of i:

- Instead of selecting the best variable to be spilt, just randomly choose it
- Instead of getting the best splitting point, just take a random one (mean of two random point)

In this way, a much faster random decision tree can be built. However the ability of classification is generally not as good as an eager decision tree. Therefore, such a random tree is used as input of bagging / boosting instead of used alone.

6. Decision Tree vs Near-Neighbor for linearly separable points on a plane

Given that the data are on a plane that can be separated by a line, it is advised to use a nearest-neighbor learning algorithm instead of decision tree.

Just imagine that the line is not vertical or horizontal: at different value of x, the data have to be separated at different level of y. It means that if a decision tree is used, a recursive, multi-layer tree is needed, while in a nearest-neighbor algorithm can do this task well on most region of the plane well.