Decision Trees

Arbres de décision

Diane Lingrand

POLYTECH NICE SOPHIA

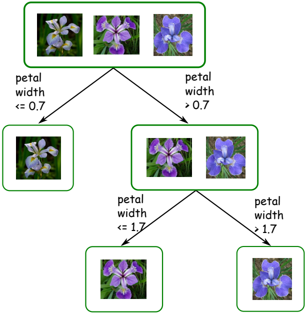
UNIVERSITÉ CÔTE D'AZUR

SI 4

2023 - 2024

Last week

- Iris dataset :
  - you built a perfect classifier for the 2 first classes
  - you built a good classifier for the 3 classes
  - both were Decision Trees !
    - binary trees
    - thresholding on one component

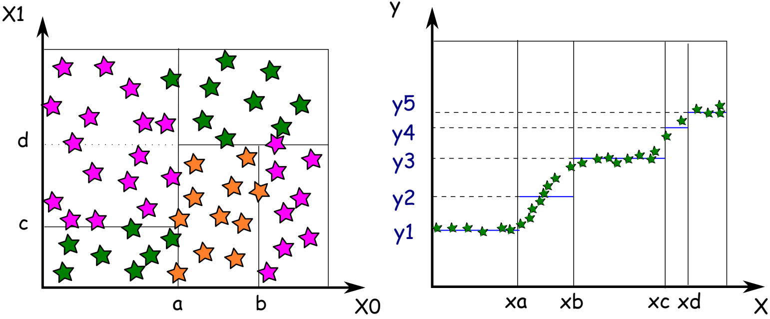


- digits dataset :
  - to much dimensions (64) to examine 2 by 2. Not enough time during the lab and not obvious to find such simple solution.
- Need to find something more automatic and perhaps more complex : this is what Decision Trees are about.

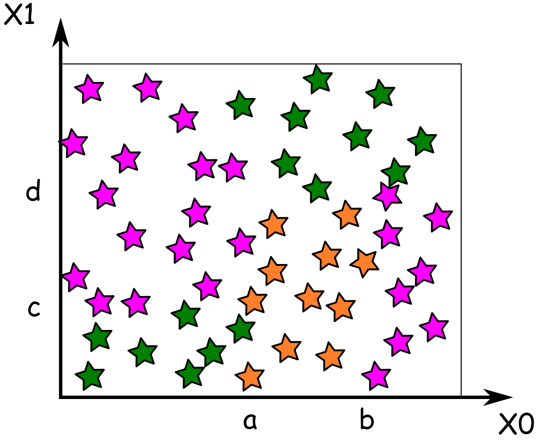
Outline

Let's start with decision trees !

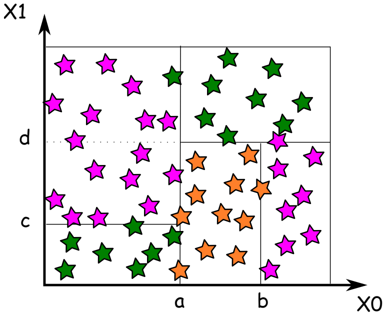
- tree of very simple decisions on the features :
  - threshold values for some of the features
- can be used both for classification and regression :

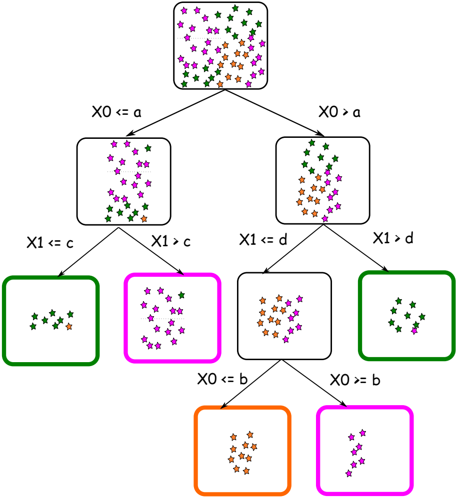


Classification (1)

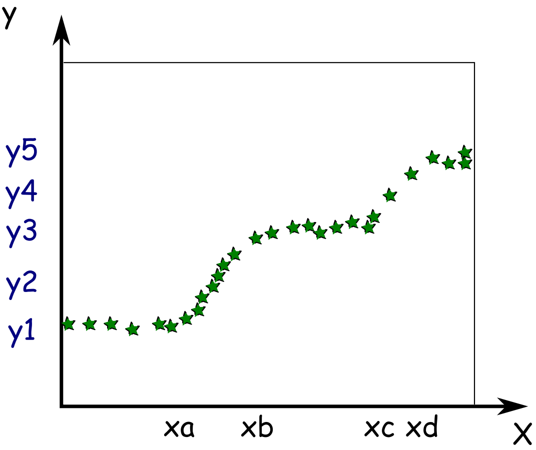


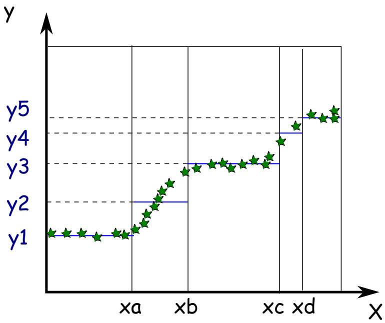
Classification (2)

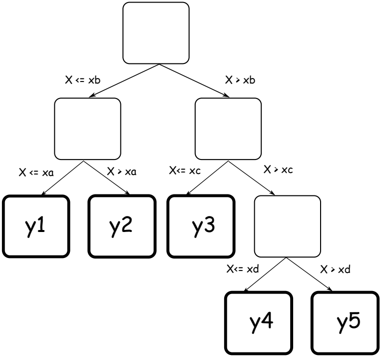




Regression (1)




y
y5
y4
y3
y2
y1
xa
xb
xc
xd
X


X <= xb
X > xb
X <= xa
X > xa
X<= xc
X > xc
y1
y2
y3
X<= xd
X > xd
y4
y5

How to decide ?

- Questions :
  - when should the tree stop growing ?

How to decide ?

- Questions :
  - when should the tree stop growing ?
    - every samples are well classified ?
    - regression value close the true value at the precision of ϵ ?
    - only one sample left in the node ?

How to decide ?

- Questions :
  - when should the tree stop growing ?
    - every samples are well classified ?
    - regression value close the true value at the precision of ϵ ?
    - only one sample left in the node ?
    - don't build very deep trees in order to avoid overfitting ! !

How to decide ?

- Questions :
  - when should the tree stop growing ?
    - every samples are well classified ?
    - regression value close the true value at the precision of ϵ ?
    - only one sample left in the node ?
    - don't build very deep trees in order to avoid overfitting ! !
  - how to select a feature and a threshold ?

How to decide ?

- Questions :
  - when should the tree stop growing ?
    - every samples are well classified ?
    - regression value close the true value at the precision of ϵ ?
    - only one sample left in the node ?
    - don't build very deep trees in order to avoid overfitting ! !
  - how to select a feature and a threshold ?
    - choose the decision that will split the data in the equally sized subsets
    - choose the decision that will lower the global error
    - need to choose a metric
    - how many tests ?

Outline

How deep ?

- fixed depth
- measure the quality of a node :
  - classification tree : continue to split if the **impurity** is too high
  - measure of impurity : $\{1, 1, 0, 1, 1\}$ is purer than $\{0, 1, 0, 1, 1\}$
  - different methods :
    - percentage of majority class : $\{1, 1, 0, 1, 1\}$ is pure class 1 at 80% while $\{0, 1, 0, 1, 1\}$ is pure class 1 at 60%
    - Entropy
    - Gini index
  - regression tree : continue to split if the cost is too high
    - mse is a metric of cost

Gini index

$$GINI(n) = \sum_{class\ \ c} p(c|n)(1 - p(c|n)) = 1 - \sum_{class\ \ c} p^2(c|n)$$

where $p(c|n)$ is the probability of class c at node n.

- Example : a node contains samples from 3 classes $\{0,0,1,1,1,2,2,2,2,2\}$:
  - class 0 : p(0) = 2/10 = 0.2
  - class 1 : p(1) = 3/10 = 0.3
  - class 2 : p(2) = 5/10 = 0.5
  - GINI = 1 - 0.04 - 0.09 - 0.25 = 0.62
- max value : 1-1/nbCl
- min value : 0

Entropy or Log Loss as measure of impurity

$$E(n) = -\sum_{class\ \ c} p(c|n)\log(p(c|n))$$

- Example : a node contains samples from 3 classes $\{0,0,1,1,1,2,2,2,2,2\}$:
  - class 0 : p(0) = 2/10 = 0.2
  - class 1 : p(1) = 3/10 = 0.3
  - class 2 : p(2) = 5/10 = 0.5
  - Log Loss = 1.03
- max value : $3\log(\mathrm{nbCl})/\mathrm{nbCl}$
- min value : 0

Outline

building a tree

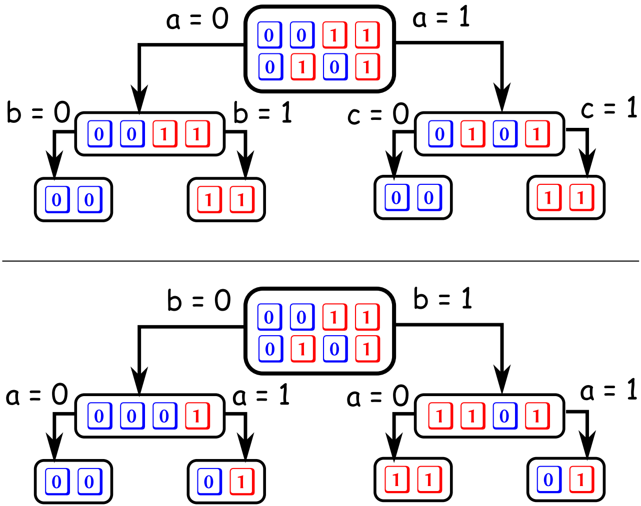
- build all the possible trees and select the best one
  - best solution
  - not possible in practice
- greedy algorithm (*algorithme glouton*)
  - best decisions are taken locally, at each split (or node)
  - compare the split candidate by a metric :

$$\frac{n_{left}}{n_{left} + n_{right}} H(nodeLeft) + \frac{n_{right}}{n_{left} + n_{right}} H(nodeRight)$$

    where H is one of the previous metrics.
  - not necessary the best solution

example

a	b	c	class
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



pruning

- during the learning process
  - in order to match the constraints (number of samples...)
- after the learning
  - in order to reduce overfitting

Conclusion

- advantages
  - easy to interpret the decision and to visualize
  - no need for preprocessing (normalisation)
    - however, it could be a good idea to reduce the dimensions of the data
  - easy to work with numerical and categorical data
  - perfom well with large dataset
  - is a way to select most important features
  - very fast inference
- drawbacks
  - unstable : a small number of data can change the tree and thus the prediction
  - easily biaised with unbalanced dataset
  - no guaranty to end with the optimal tree
  - not the more precise ml algorithm (only thresholding components)
  - overfitting

Outline

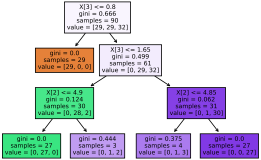
scikit-learn implementation

- Documentation available :
  https://scikit-learn.org/stable/modules/tree.html
  - implements CART algorithm
- Code sample :

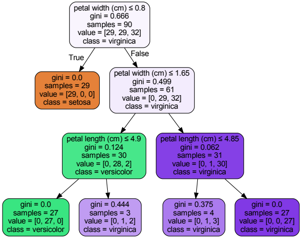
```
from sklearn import tree
myTree = tree.DecisionTreeClassifier()
myTree.fit(X_train, y_train)
ypred = mTree.predict(X_test)
# and then compute the metrics ...
```

- Useful parameters :
  - max_depth
  - min_samples_leaf or min_samples_split
  - other parameters description at https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier

- using `plot_tree` (graphical output) :
  `https://scikit-learn.org/stable/modules/generated/sklearn.tree.plot_tree.html#sklearn.tree.plot_tree`
- using `export_graphviz` (more graphical options) : `https://scikit-learn.org/stable/modules/generated/sklearn.tree.export_graphviz.html#sklearn.tree.export_graphviz`
- using `export_text` (textual output) : `https://scikit-learn.org/stable/modules/generated/sklearn.tree.export_text.html#sklearn.tree.export_text`



`plot_tree`



`export_graphviz`

```
|--- petal width (cm) <= 0.80
|   |--- class: 0
|--- petal width (cm) >  0.80
|   |--- petal width (cm) <= 1.65
|   |   |--- petal length (cm) <= 5.00
|   |   |   |--- class: 1
|   |   |--- petal length (cm) >  5.00
|   |   |   |--- class: 2
|   |--- petal width (cm) >  1.65
|   |   |--- petal length (cm) <= 4.85
|   |   |   |--- class: 2
|   |   |--- petal length (cm) >  4.85
|   |   |   |--- class: 2
```

`export_text`

Today's lab

- Iris dataset : which tree is computed ? Can you perform better than last week ?
- digits dataset : same questions
- Play with the hyperparameters of the decision tree algorithm.
  - for each value of hyperparameters :
    - learn the tree with the **train set**
    - compute metrics with the **validation set**
  - choose the hyperparameters according to the best metrics on the **validation set**
  - compute and publish metrics with the **test set**
- You can continue on the same notebook or write a new one.