MACHINE LEARNING FOR DATA MINING
LECTURE 7: Decision Tree

DECISION TREE CLASSIFICATION

Siamak Sarmady (UUT, UU)
Viktor Lavrenko (Uni. Of Edinburgh)
Alexandre Ihler (UCI)

Concept

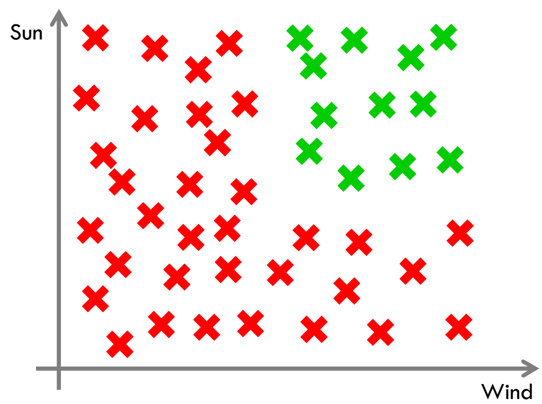
Decision Tree Classifier

- These classifiers go back to decades (used in Rule Based Expert systems as well as other applications) and they are very popular.
- They are extremely robust.
- They have interesting decision boundaries and can do non-linear classification.
- They are intuitive. You can look at the results and understand how the model has been built and works.

- Trick
  - Just like SVM that uses Kernel trick to do non-linear classification using linear boundaries, decision trees use linear decision boundaries (many of them) to build a non-linear decision boundary.

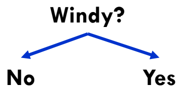
Decision Trees - Example

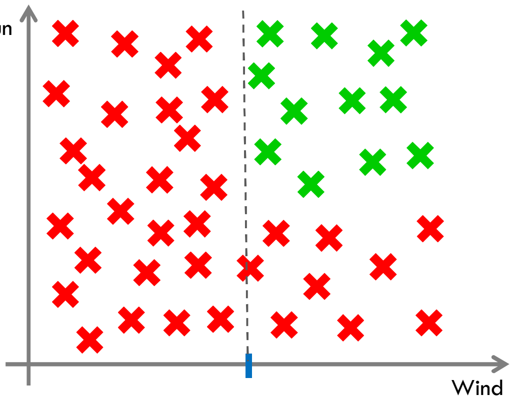
- John likes to windsurf. But to do that he needs wind and he likes the weather to be sunny when he windsurfs.
- We take his surrfing log, find the weather status on each day and put the data points on a graph.
- **Question:** Is this data linearly separable?
  - No, you cannot find a line to separate these classes.



Decision Trees - Example

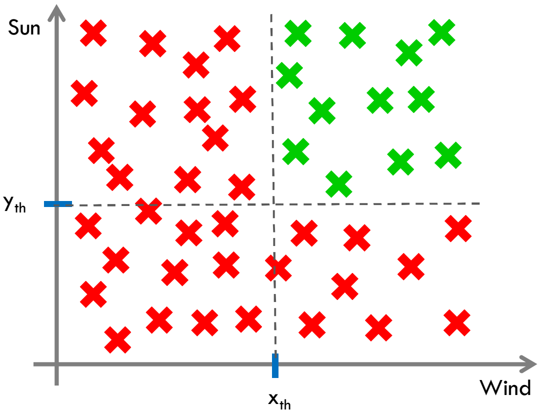
- Decision trees allow you to ask multiple linear questions one after another:
- Is it windy?
- We can find some kind of threshhold that can build a partial boundary for the classes.
- This is equal to answering the following question. In fact the question represents a linear and (partial) class boundary (being windy or not)

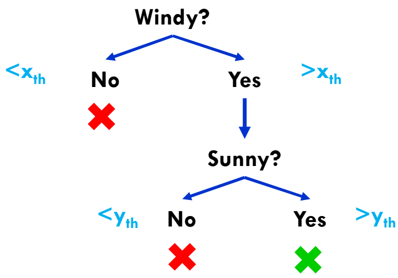




Decision Trees - Example

- One of the two branches (No Branch) does not contain any positive (green) data. If it's not windy, all the data points are negative.
- In the case of "Yes branch" we need to ask a further question.

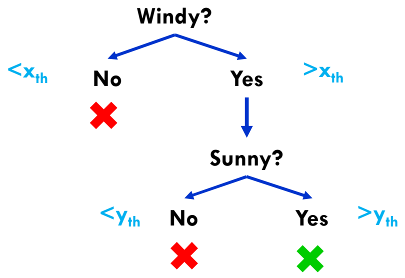




- The algorithm takes the data, finds suitable thresholds and builds a tree like the above.

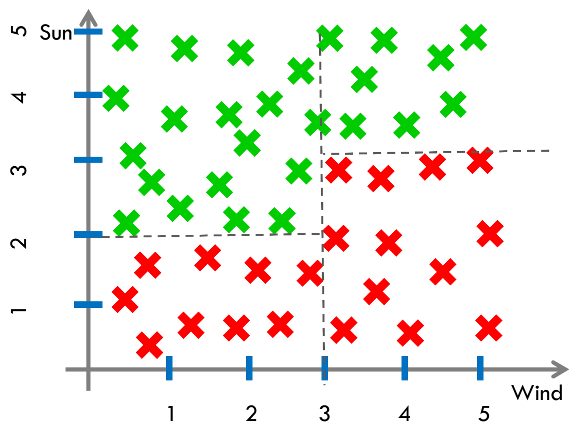
Decision Trees - Example

- The algorithm takes the data, finds suitable thresholds and builds a tree.
- After the decision tree (and the threshhold for each branch) is known, the tree can be used to classify data.



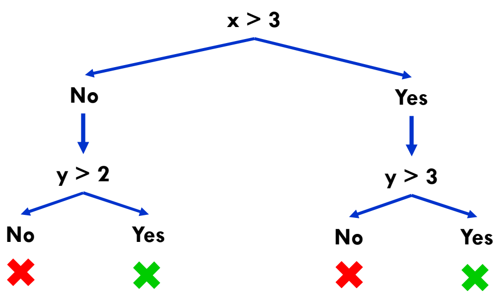
Decision Trees – Example 2

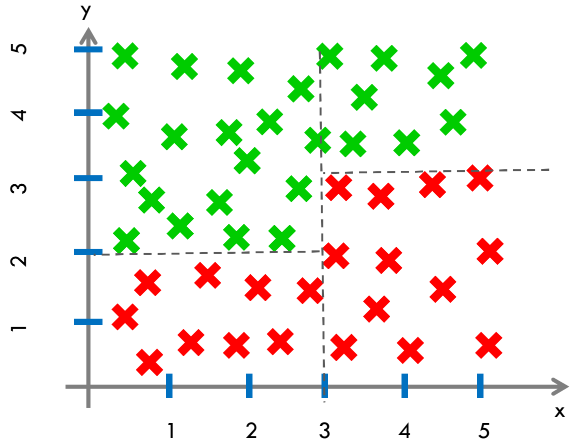
- Let's consider a complicated example.
- Which x_{th} is the best threshold for the horizontal axis?
  - X=3 can build a partial class boundary.
  - However it cuts both classes into two.
- We need more boundaries to do a proper classification.
- Every straight line matches a binary question…
- Note: we selected X as the first attribute that the space is divided on it… This decision should be done in a more formal and robust way



Decision Trees – Example 2

- So we can draw the following decision tree for the provided training data set.





- Note: that this is one of the decision trees that we can draw for this data.

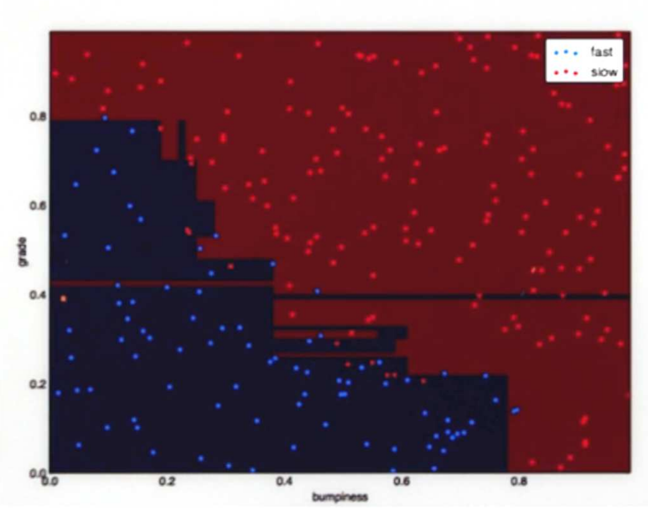
Decision Tree using Scikit Learn

Using SkLearn – Program 1 (Data from Array)

```
import numpy as np

features_train = np.array([[-1, -1], [-2, -1], [-3, -2], [1, 1], [2, 1],[3,3]])
labels_train   = np.array([1, 1, 1, 2, 2, 2])
features_test  = np.array([[-2, -2], [-2, -3], [2, 3], [1, 2]])
labels_test    = np.array([1, 1, 1, 2])

from sklearn import tree

clf = tree.DecisionTreeClassifier()
clf.fit(features_train, labels_train)
pred = clf.predict(features_test)

print "Test labels:      ", labels_test
print "Predicted labels: ", pred

from sklearn.metrics import accuracy_score
print "Accuracy:         ", accuracy_score(pred, labels_test)

print "\nPredicted label for ", [-0.8, -1] ," is ", (clf.predict([[-0.8, -1]]))
```

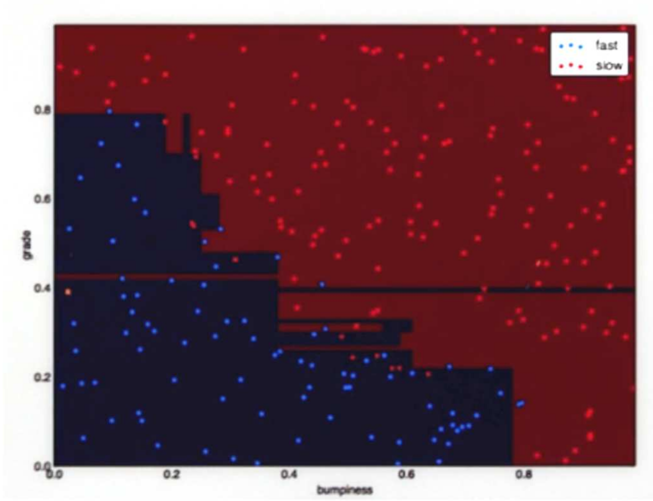
Decision Tree – The boundary

- The figure visualizes the class boundaries a DT classifier has built.
- As you see the DT algorithm has been able to build a non-linear boundary using linear boundaries.
- The graph shows the test data. Training data points are NOT shown.
- The boundary has shortcomings. The narrow areas are the result of over-fitting to the training data
- The narrow blue areas have been learned from blue points in the training data (and the narrow red areas from red data points)



Decision Tree – Overfitting

- The over-fitting has happened because the algorithm has built a tree that is too deep (i.e. builds more sub-regions)
- If we want to generalize better and avoid over-fitting, we should limit the depth that the algorithm gets into (i.e. build a tree with limited levels).

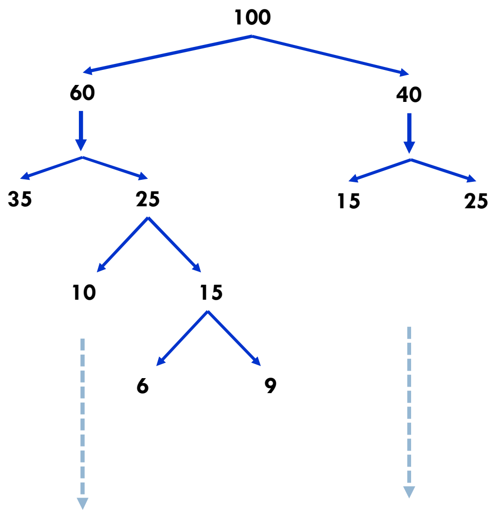


Using SkLearn – Program 1 (Data from Array)

- The function we used from sklearn provides several adjustable parameters:

```
class sklearn.tree.DecisionTreeClassifier(criterion='gini',
splitter='best', max_depth=None, min_samples_split=2,
min_samples_leaf=1, min_weight_fraction_leaf=0.0,
max_features=None, random_state=None, max_leaf_nodes=None,
class_weight=None, presort=False)
```

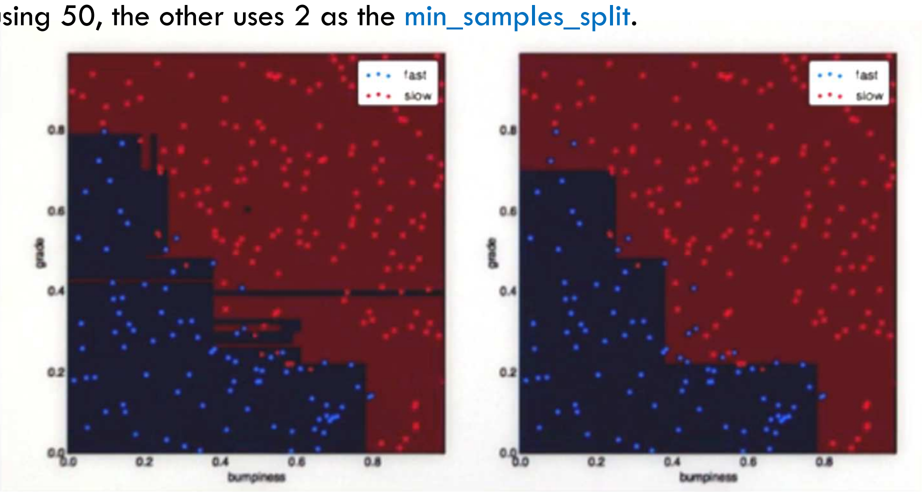
- Most of the above parameters will affect over fitting and accuracy. We test a few. Mostly we want to control how far the splitting happens in different branches.
  - **max_depth:** determines the maximum depth of the tree
  - **min_samples_split:** determines when the algorithm should stop splitting an area (using new rules)
    - If we set it to 6, which branch will not be split anymore?



Decision Tree – Overfitting

- Guess which one of the following graphs has used a larger min_samples_split parameter? (i.e. minimum samples to allow split)
- One is using 50, the other uses 2 as the min_samples_split.

90.8%



91.2%

Entropy

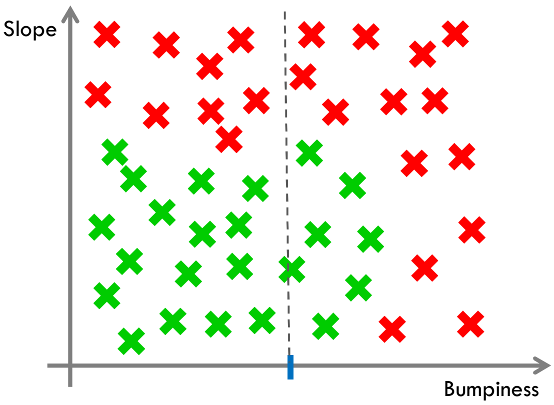
How the decision boundaries are determined

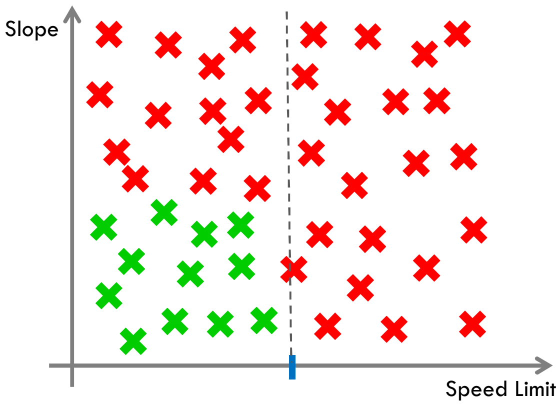
Entropy

- The decision tree algorithm uses a concept called Entropy to decide on what attributes and where (what threshold) it should split the data.
- **Entropy:** measure of impurity in a bunch of data (training data)
  - It measures the opposite of purity…

Entropy

- If we are supposed to make a decision tree by either splitting speed limit attribute range into two or the bumpiness level into two, which one would cause more impurity (entropy)?
- Splitting the speed limit attribute would give lower entropy (in the right hand side box) and therefore is better. We recursively select splits that are as pure as possible.





Entropy

- The mathematical formula of entropy is as follows:

$$Entropy = \sum_i -P_i \log_2(P_i)$$

  - P_i is the fraction of consistent examples in a given class i.
  - We sum up the number over all classes (labels) and get the total entropy
  - In each split:
    - All of the same class → Entropy=0
    - Data are 50-50 of two different classes → Entropy=1 (maximum)

Information Gain

- After we have done splits, the entropy decreases in the splits.
- If we calculate the weighted average of entropy in the split data, we can find how much the entropy has decreased.
- **Information Gain:** decrease of entropy means more organized understanding of the structure of the data and we call it information gain

$$Information\ Gain = Entropy(parent) - [weighted\ average]Entropy(Child\ splits)$$

  - Decision tree checks different splits in each step and performs the one which gives higher information gain.

Entropy - Example

- Assume we have the following data:

Slope	Bumpiness	Speed-Limit (activated)	Speed (Classes)
Steep	Bumpy	Yes	Slow
Steep	Smooth	Yes	Slow
Flat	Bumpy	No	Fast
Steep	Smooth	No	Fast

SSFF

- What is the initial entropy in all classes before we do any split, i.e. in an unsplit area there are 2 series of data with 50-50 distribution (i.e. a measure of how we understand the data):
- $\text{entropy slow} = (-\frac{2}{4}\log_2\frac{2}{4})$

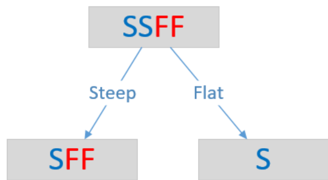
  $\text{entropy fast} = (-\frac{2}{4}\log_2\frac{2}{4})$

  $\text{entropy} = \sum_i -P_i\log_2(P_i) = (-\frac{2}{4}\,.-1) + (-\frac{2}{4}\,.-1) = 1.0$ (i.e. very bad)

Entropy - Example

- Now if we split data into two groups based on the attribute "Slope" what is the entropy:

Slope	Bumpiness	Speed-Limit (activated)	Speed (Classes)
Steep	Bumpy	Yes	Slow
Steep	Smooth	Yes	Slow
Flat	Bumpy	No	Fast
Steep	Smooth	No	Fast



- What is the entropy and information gain if we split the data based on "slope" attribute.

  $\text{entropy slow} = (-\frac{2}{3}\log_2\frac{2}{3})$

  $\text{entropy fast} = (-\frac{1}{3}\log_2\frac{1}{3})$

  $$\text{entropy of steep branch} = \sum_i -P_i\log_2(P_i) = (-\frac{2}{3}\,.\text{-}0.584) + (-\frac{1}{3}\,.\text{-}1.584) = 0.9182$$

  $$\text{entropy of flat branch} = \sum_i -P_i\log_2(P_i) = (-1\,.\,0) = 0$$

Entropy - Example

- So entropies are (weights are the ratio of data in each branch):

  $$[weighted\ average]Entropy(Child\ splits) = \left(\frac{3}{4}\ 0.9184\right) + \left(\frac{1}{4}\ 0.0\right) = 0.6888$$

  $$Entropy(Parent) = 1.0$$ (it was a 50-50 split between classes)

- **Information gain:**

  $$Information\ Gain = 1 - 0.6888 = 0.3112$$