Decision Trees

Houcine Senoussi

April 12, 2016

What is it about ?

- Supervised learning also called classification
- Several types of SL methods, among them 'Decision Tree'.

Basic concepts

- Given a set D of data records, called **examples**.
  - D is called **training set** or **training data**.
- Each data record is described using attributes $\{A_1, A_2, ..., A_p\}$.
- Each attribute A_p has n_p possible valeurs v_1, ..., v_{np}.
- There is a special attribute C, called **class** attribute.
- Attribute C has m possible values c_1, ..., c_m.
- Objective : Produce a classification/prediction function using a learning algorithm.
  - Prediction : predict class values of the future data.
- In this chapter this function is in form of a decision tree.
- After this function is built, it is evaluated using a set of test data (unseen instances).

Basic concepts-2

- Test set :
  - The examples in the test data also have class labels.
    - Testing : comparing given class labels with those predicted by the function.
  - Usually available data is split into two disjoint sets : the training set and the test set.
- Accuracy $= \frac{Number-of-correct-classifications}{Total-number-of-test-cases}$.

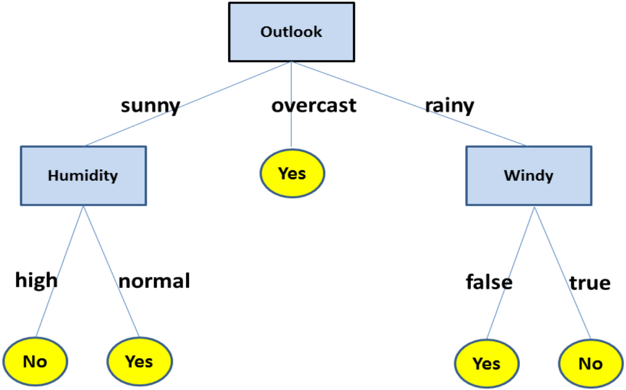
Example

Id	Outlook	Temperature	Humidity	Windy	PlayTennis
1	sunny	hot	high	false	**No**
2	sunny	hot	high	true	**No**
3	overcast	hot	high	false	**Yes**
4	rainy	mild	high	false	**yes**
5	rainy	cool	normal	false	**Yes**
6	rainy	cool	normal	true	**No**
7	overcast	cool	normal	true	**Yes**
8	sunny	mild	high	false	**No**
9	sunny	cool	normal	false	**Yes**
10	rainy	mild	normal	false	**Yes**
11	sunny	mild	normal	true	**Yes**
12	overcast	mild	high	true	**Yes**
13	overcast	hot	normal	false	**Yes**

Example - 2

- Attributes :
  - Outlook : sunny, overcast, rainy.
  - Temperature : hot, mild, cool.
  - Humidity : high, normal.
  - Windy : true, false.
- Class Attribute : PlayTennis. 2 values : Yes, No.
- The following decision tree corresponds to this data.

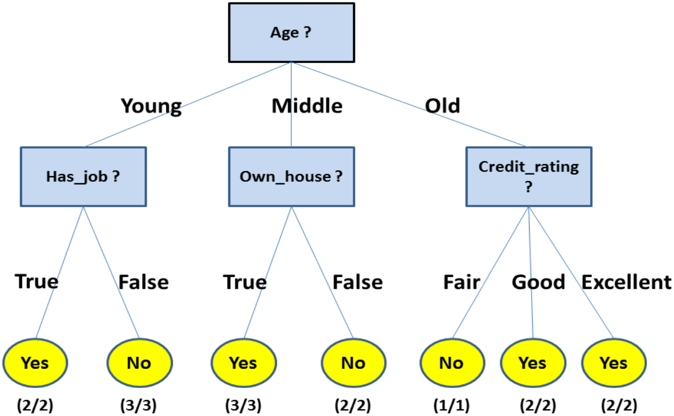
Example-3


Outlook
sunny
overcast
rainy
Humidity
Yes
Windy
high
normal
false
true
No
Yes
Yes
No

What is a Decision Tree ?

- A decision tree is a hierarchical decomposition of a data set.
- At each each internal node, a condition is used to divide a subset according to instances' properties.
  - Each (internal) node is labeled by an attribute.
  - Each edge (branch) from such a node corresponds to a value of the attribute.
- Each leaf indicates a class.
- To use the decision tree to classify an instance :
  1. We traverse the tree top-down according to the attribute values of the given instance.
  2. until we reach a leaf node : The class of the leaf is the predicted class of the instance.

What is a Decision Tree ?-Example



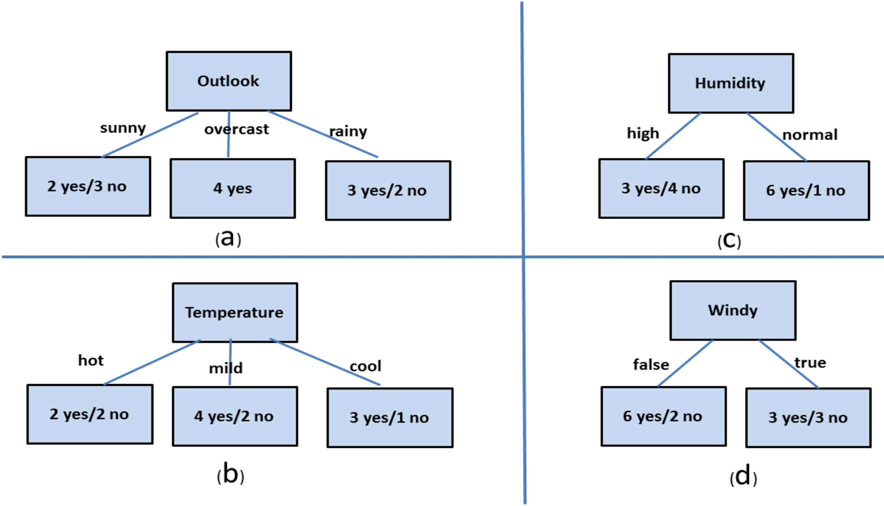
Constructing Decision Trees

1. Given a set D of examples described by the attributesA_1, ..., A_p and belonging the classes C_1, ..., C_m.
2. First, **select** an attribute to place at the root node.
3. Make one branch for each possible value of this attribute.
4. Each branch ends with a node corresponding to a subset of D.
5. For each node :
    1. If all elements of the set corresponding to it belong to the same class C_i, then make it a leaf labeled with C_i.
    2. else repeat recursively steps 2-5.

'Purity' and Information gain

- The question is : given a node of the decision tree (and the set of examples corresponding to it) :
  - How to determine the attribute to place at this node (i.e. to split on the set of examples) ?
- Let us consider again our example.
  - There are 4 possibilities/attributes to split on the set D.
- Which is the best choice ?
- The criterion for the choice is as follows :
  - Since we seek small trees, we would like to reach leaves as soon as possible.
  - Therefore, among these possibilities we will choose the one that produce the '**purest**' subsets.

Example-4


Outlook
sunny
overcast
rainy
2 yes/3 no
4 yes
3 yes/2 no
(a)
Humidity
high
normal
3 yes/4 no
6 yes/1 no
(c)
Temperature
hot
mild
cool
2 yes/2 no
4 yes/2 no
3 yes/1 no
(b)
Windy
false
true
6 yes/2 no
3 yes/3 no
(d)

'Purity' and Information gain-2

- 'Purity' of a subset depends on the number of yes and no's that it contains :
  - The maximal purity is reached by a set containing only yes or only no's.
  - The maximal impurity is reached by a set containing the same number of yes and no's.
- There is a function that can caracterise purity : the **entropy**.

Entropy

Definition

Given a set D and n classes $(C_1,\ldots,C_n)$. The entropy of D is defined by :

$$\mathrm{Ent(D)} = -\sum_{i=1}^{n}(p_i * logp_i)$$

where

- $p_i = \frac{number\ \ of\ \ elements\ \ of\ \ D\ \ belonging\ \ to\ \ C_i}{total\ \ number\ \ of\ \ elements\ \ of\ \ D}$

Entropy-2

Let us take some examples :

- If we have n classes and for each class $p_i = \frac{1}{n}$ (all the classes have the same number of elements). then :
  - Ent(D) $= -n * \frac{1}{n} * log(\frac{1}{n}) = logn$
  - Ent(D) $= 1$ in the case when $n = 2$ (two classes {True, False}, {Yes, No}, ...).
- If we have $p_k = 1$ for some value k (all the elements belong to the same class). then :
  - Ent(D) $= 0$

Entropy-Example

- In our example we have :
  - two classes : *Yes* and *No*.
  - The number of examples belonging to the class *Yes* is 9 . Therofore, $p_{yes} = \frac{9}{14}$.
  - The number of examples belonging to the class *No* is 5 . Therofore, $p_{yes} = \frac{5}{14}$.
  - The entropy E is equal to $-\frac{9}{14} log \frac{9}{14} - \frac{5}{14} log \frac{5}{14} = 0.940$.

Entropy-Information gain

- Since our aim is to improve purity, we will choose the attribute for which we will have the highest value of
  - $IG = Entropy\_before\_spiliting\_on$ - $Entropy\_after\_spiliting\_on$.
- This number is called **Information Gain**.

Entropy-Example(2)

- If we use attribute 'outlook' to split on D, we obtain :
  - a subset D_1 : Size = 5, Entropy = 0.97.
  - a subset D_2 : Size = 4, Entropy = 0.
  - a subset D_3 : Size = 5, Entropy = 0.97.
- Therefore, the 'new entropy' is equal to :
  - $E_{outlook} = \frac{5}{14}0.97 + \frac{4}{14}0 + \frac{5}{14}0.97 = 0.69$.
  - Using this attribute would improve purity by $IG_{outlook} = E$ - $E_{outlook}$ = 0.94 - 0.69 = 0.25.

Entropy-Example(3)

- If we use attribute 'Temperature' to split on D, we obtain :
  - a subset D_1 : Size = 4, Entropy = 1.
  - a subset D_2 : Size = 6, Entropy = 0.92.
  - a subset D_3 : Size = 4, Entropy = 0.81.
- Therefore, the 'new entropy' is equal to :
  - $E_{temperature} = \frac{4}{14}1 + \frac{6}{14}0.92 + \frac{4}{14}0.81 = 0.91$.
  - Using this attribute would improve purity by $IG_{temperature} = E$ - $E_{temperature} = 0.94$ - $0.91 = 0.03$.

Entropy-Example(4)

- If we use attribute 'Humidity' to split on D, we obtain :
  - $E_{humidity} = 0.79$.
  - Using this attribute would improve purity by $IG_{humidity} = E$ - $E_{humidity} = 0.94$ - $0.79 = 0.15$.
- If we use attribute 'Windy' to split on D, we obtain :
  - $E_{Windy} = 0.89$.
  - Using this attribute would improve purity by $IG_{Windy} = E$ - $E_{Windy} = 0.94$ - $0.89 = 0.05$.

Algorithm

- Decision-Tree(Input : D, A; Output T)
  1. **if** D contains only examples of the class c_i **then** $T = \{$one leaf labeled with $c_i\}$.
  2. **elseif** $A = \emptyset$ **then** $T = \{$one leaf labeled with the most frequent class $c_M\}$.
  3. **else**
     1. Decision-Tree-2(D, A, T).
  4. **endif**

Algorithm-2

- Decision-Tree-2(Input : D, A; Output T)
  1. Compute p_0 = Entropy(D).
  2. **foreach** attribute A_i

     compute p_i = Entropy-After-Partition(D, A_i).

     endforeach.
  3. Select the attribute A_g that maximizes $p_i - p_0$.
  4. Make T a decision node on A_g.
  5. Partition D into m subsets $D_1, \ldots, D_m$ based on the m values of A_g.
  6. **foreach** $D_j \in \{D_1, \ldots, D_m$ **do**
     1. **if** $D_j \neq \emptyset$ **then**

        Create a branch node T_j as a child node of T.

        Decision-Tree(D_j, A-$\{A_g\}$, T_j)
     2. **endif**.
  7. **endforeach**.

Algorithm-2

- Function Entropy-After-Partition : Input D, A_i; Returns p_i.
  1. Let v_1, ..., v_p the possible values of A_i.
  2. If we use A_i to partition D, we will divide D into p subsets D_1, ..., D_p.
  3. $p_i = \sum_{j=1}^{p} \frac{|D_j|}{|D|}.entropy(D_j)$.
  4. Return p_i.

Summary

- Describing a SL problem.
- Using a decision tree to classify data.
- Constructing a decision tree.
  - Entropy and Information Gain.

Exercices

- Use Weka to define a decision tree for :
  1. the weather problem.
  2. the loan problem.

References

- Liu B. Web Data Mining. Springer. 2007, 532 pages.
- Witten I. H., Frank E., Hall M. A. Data Mining. Morgan Kaufmann Publishers. 2011, 628 pages.