Applied Machine Learning

Lecture 10
Decision Tree

Ekarat Rattagan, Ph.D.

Outline

1. **Introduction**
2. **Classification And Regression Trees (CART)**
3. **Iterative Dichotomiser 3 (ID3)**

Introduction

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
5	6	5.4	3.9	1.7	0.4	Iris-setosa
6	7	4.6	3.4	1.4	0.3	Iris-setosa
7	8	5.0	3.4	1.5	0.2	Iris-setosa
8	9	4.4	2.9	1.4	0.2	Iris-setosa
9	10	4.9	3.1	1.5	0.1	Iris-setosa

Introduction

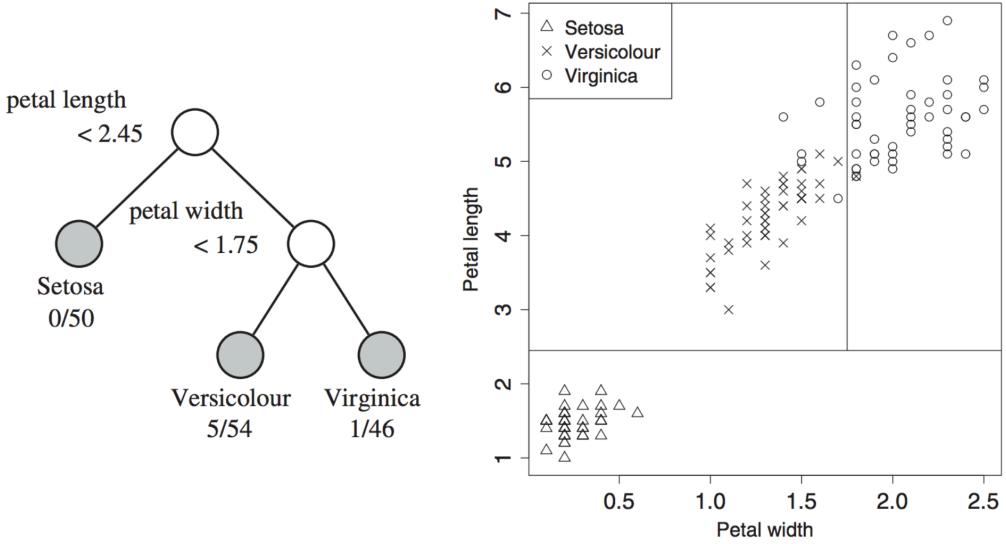


Figure 1. *Classification tree model for iris data. At each intermediate node, an observation goes to the left child node if and only if the stated condition is true. The pair of numbers beneath each terminal node gives the number misclassified and the node sample size.*

Introduction

G(outlook)

Calculate weighted sum of gini indexes for outlook features

sun over Rain

$\frac{5}{14} \times 0.48 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.48$

$= 0.342$

G(value of node)

$G(\text{sunny}) = 1 - \left[\left(\frac{2}{5}\right)^2 + \left(\frac{3}{5}\right)^2\right]$

$= 0.48$

	0.342	9	9	9	
Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

→ gini of Temperature

Humidity

Wind

จำนวน class y_i, $N=2$

$1 - \sum_{i=1}^{N} P^2\left(\frac{\#i}{\#t}\right)$

จำนวนของ value ที่ class i

จำนวนของ value ทั้ง

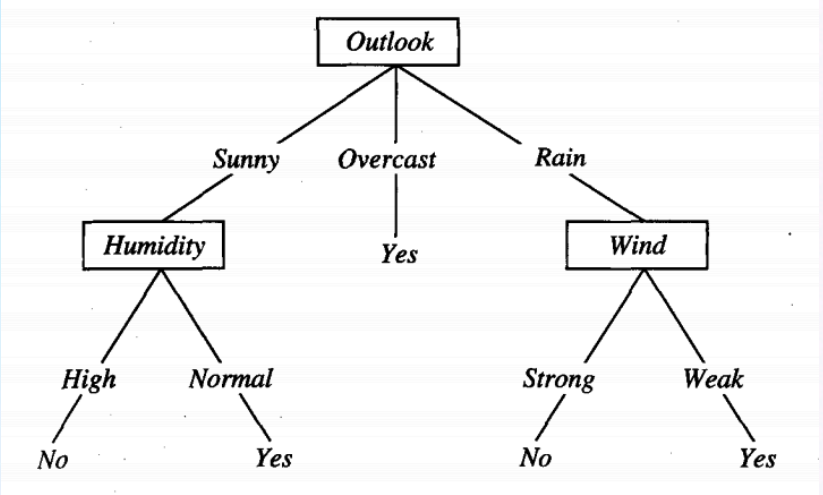
$G(\text{Rain}) = 1 - \left[\left(\frac{3}{5}\right)^2 + \left(\frac{2}{5}\right)^2\right] = 0.48$

G(overcast)

$= 1 - \left[\left(\frac{4}{4}\right)^2 + \left(\frac{0}{4}\right)^2\right] = 0$

Credit: Tom M. Mitchell

Introduction



A decision tree

Each internal node: attribute x_j
Each branch from a node: selects one value for x_j
Each leaf node: Y

This decision tree is equivalent to:

if $(Outlook = Sunny) \wedge (Humidity = Normal)$ then Yes;

if $(Outlook = Overcast)$ then Yes;

if $(Outlook = Rain) \wedge (Wind = Weak)$ then Yes;

Credit: Tom M. Mitchell

CART

- A binary tree classifier
- Gini impurity (index) is a measure of the homogeneity (or "purity") of the nodes. If all data at one node belong to the same class then this node is considered "pure". So by minimising the Gini impurity the decision tree finds the features the separate the data best [quora]
  - Gini index = $1 - \sum_{i=1}^{n} p^2(i \mid t)$, where $p(i \mid t)$ is the proportion of class i observations in node t.

Example

- https://github.com/ekaratnida/applied-machine-learning/blob/master/Week10-desicion-tree/running.ipynb

ID3 Algorithm

ID3(*Examples*, *Target_attribute*, *Attributes*)

Examples are the training examples. Target_attribute is the attribute whose value is to be predicted by the tree. Attributes is a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given Examples.

- Create a *Root* node for the tree
- If all *Examples* are positive, Return the single-node tree *Root*, with label = +
- If all *Examples* are negative, Return the single-node tree *Root*, with label = −
- If *Attributes* is empty, Return the single-node tree *Root*, with label = most common value of *Target_attribute* in *Examples*
- Otherwise Begin
  - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*
  - The decision attribute for *Root* $\leftarrow A$
  - For each possible value, v_i, of A,
    - Add a new tree branch below *Root*, corresponding to the test $A = v_i$
    - Let $Examples_{v_i}$ be the subset of *Examples* that have value v_i for A
    - If $Examples_{v_i}$ is empty
      - Then below this new branch add a leaf node with label = most common value of *Target_attribute* in *Examples*
      - Else below this new branch add the subtree ID3($Examples_{v_i}$, *Target_attribute*, *Attributes* − {A})
- End
- Return *Root*

Credit: Tom M. Mitchell

Information Gain VS Gini

- **central choice**: Which attribute classifies the examples best?
- ID3 uses the **information gain**

  outlook

  - statistical measure that indicates how well a given attribute separates the training examples according to their target classification
  - ID3 uses this **information gain measure** to select among the candidate attributes at each step while growing the tree.

Entropy

$-\frac{5}{7}\log_2\frac{5}{7} - \frac{2}{7}\log_2\frac{2}{7}$

- Entropy is a measure of the impurity in a collection of training examples.

$$Entropy(S) \equiv -p_{\oplus}\log_2 p_{\oplus} - p_{\ominus}\log_2 p_{\ominus}$$

tennis

S: a collection containing positive and negative examples of some target concept.

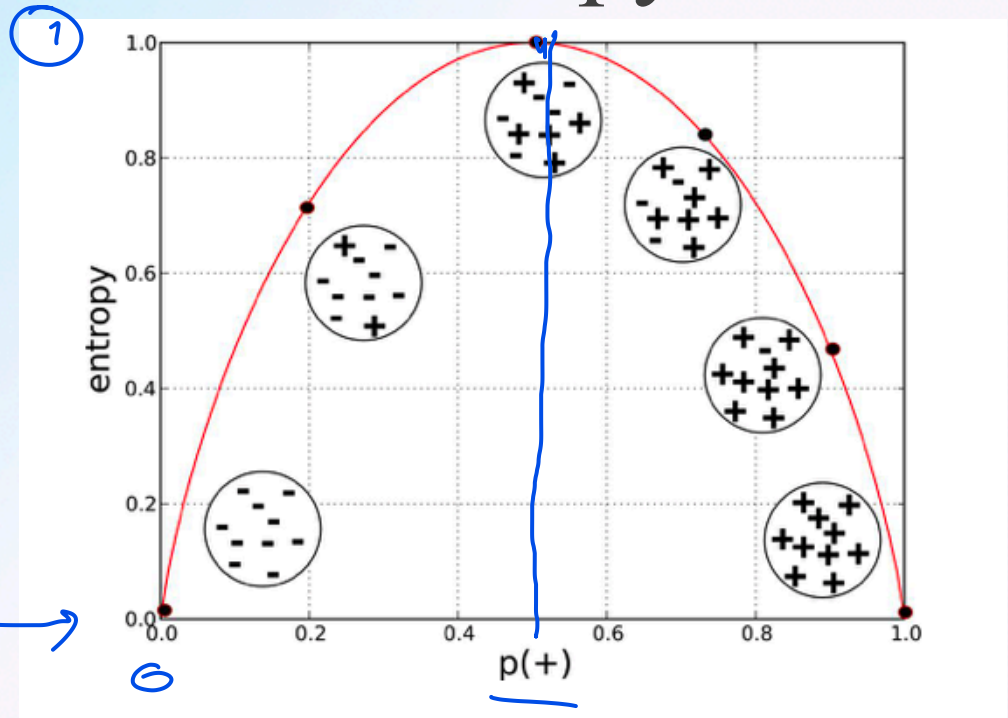
$p_{\oplus}$ is the proportion of positive examples in S.

$p_{\ominus}$ is the proportion of negative examples in S.

14

$$Entropy([9+, 5-]) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14)$$
$$= 0.940$$

Entropy



Provost, Foster; Fawcett, Tom. Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking

Information Gain

- **central choice**: Which attribute classifies the examples best?
- ID3 uses the **information gain**
  - The expected reduction in entropy caused by partitioning the examples according to this attribute
  - $$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (3.4)$$

where $Values(A)$ is the set of all possible values for attribute A, and S_v is the subset of S for which attribute A has value v (i.e., $S_v = \{s \in S | A(s) = v\}$). Note the first term in Equation (3.4) is just the entropy of the original collection S, and the second term is the expected value of the entropy after S is partitioned using attribute A. The expected entropy described by this second term is simply the sum of the entropies of each subset S_v, weighted by the fraction of examples $\frac{|S_v|}{|S|}$ that belong to S_v. $Gain(S, A)$ is therefore the expected reduction in entropy caused by knowing the value of attribute A.

ID3

$\text{Gain}(S, \text{Humidity})$

$= \text{Entropy}(S)$

$0.940 - \frac{\text{High}}{\#S} \times \text{Ent}(3+,4-)$

Which attribute is the best classifier?

Gain (S, Humidity)

= .940 - (7/14).985 - (7/14).592
= .151

Gain (S, Wind)

= .940 - (8/14).811 - (6/14)1.0
= .048

$$0.940 - \frac{7}{14} \times \underbrace{\left(-\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7}\right)}_{0.985} - \frac{7}{14} \times \underbrace{\left(-\frac{6}{7}\log_2\frac{6}{7} - \frac{1}{7}\log_2\frac{1}{7}\right)}_{.592}$$

Credit: Tom M. Mitchell

ID3

- informations gains for the four attributes:

  $Gain(S, Outlook) = 0.246$

  $Gain(S, Humidity) = 0.151$

  $Gain(S, Wind) = 0.048$

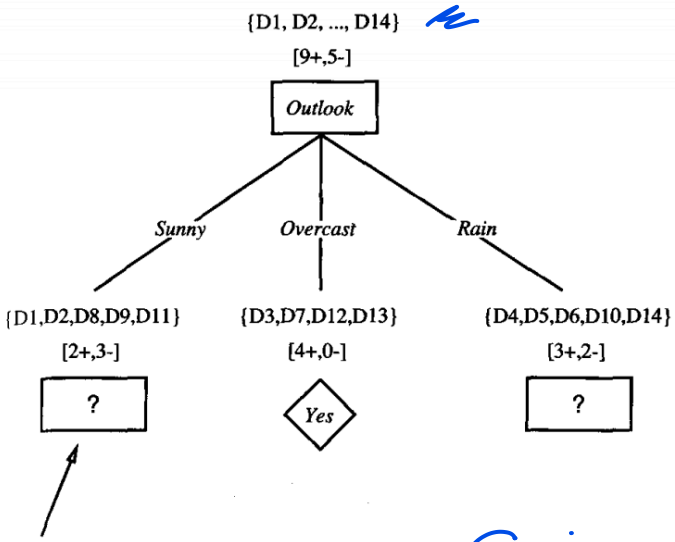
  $Gain(S, Temperature) = 0.029$

$\Rightarrow$ $Outlook$ is selected as best classifier and is therefore $Root$ of the tree

$\Rightarrow$ now branches are created below the root for each possible value

  - because every example for which $Outlook = Overcast$ is positive, this node becomes a leaf node with the classification Yes
  - the other descendants are still ambiguous
  - hence, the decision tree has to be further elaborated below these nodes

ID3



Which attribute should be tested here?

Gain (S_{sunny}, Humidity)

S_{sunny} = {D1,D2,D8,D9,D11}

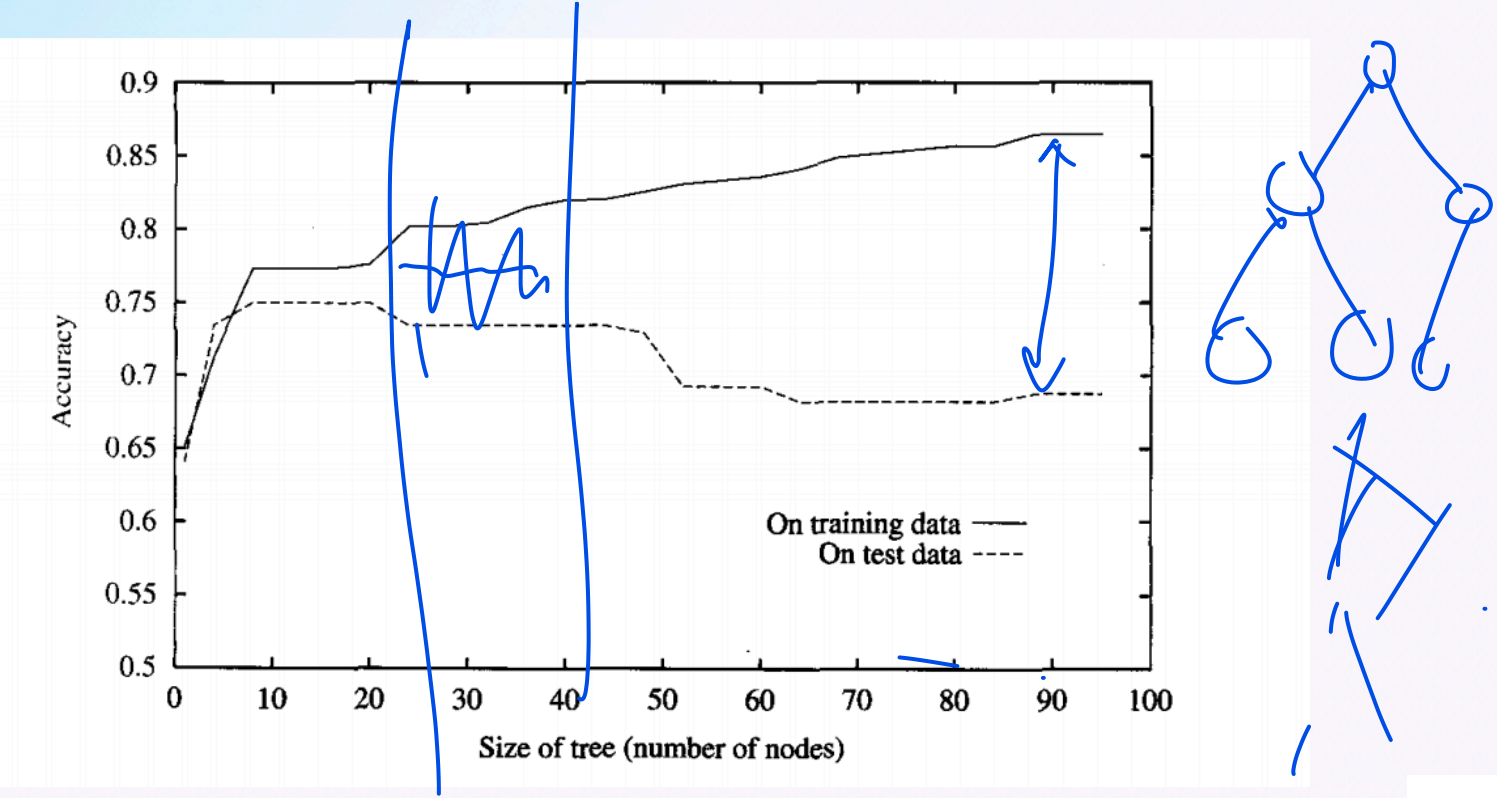
Gain (S_{sunny}, *Humidity*) = .970 - (3/5) 0.0 - (2/5) 0.0 = .970

Gain (S_{sunny}, *Temperature*) = .970 - (2/5) 0.0 - (2/5) 1.0 - (1/5) 0.0 = .570

Gain (S_{sunny}, *Wind*) = .970 - (2/5) 1.0 - (3/5) .918 = .019

Credit: Tom M. Mitchell

Issue: Overfitting

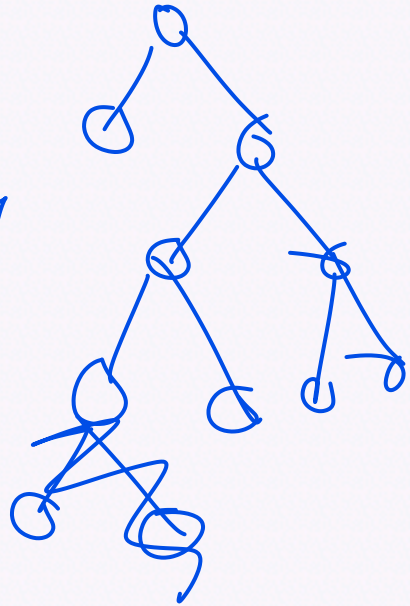


Solution: Overfitting

1. Limit the number of iterations of ID3
   - Leading to a tree with a bounded number of nodes
2. Pruning (after it is built)

① depth

② pruning

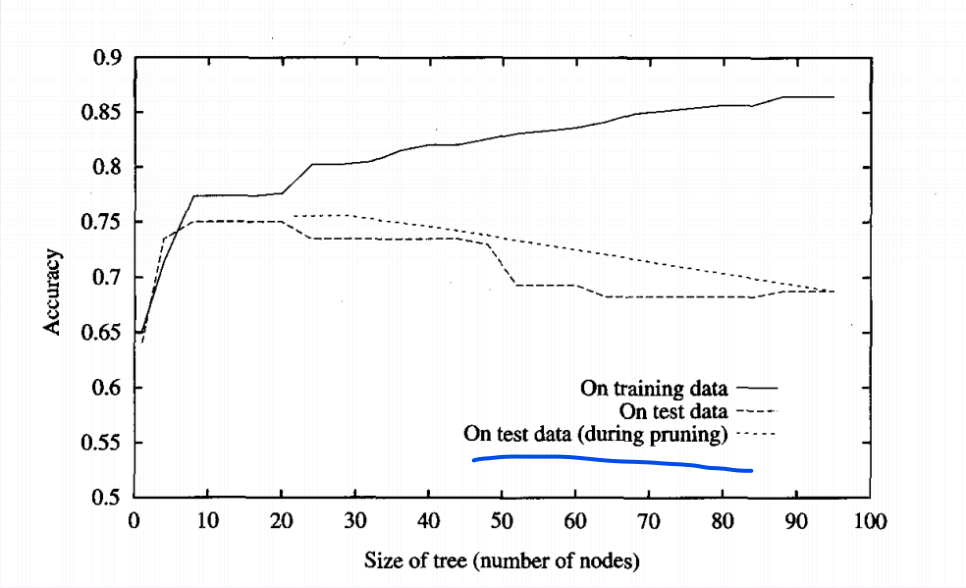


```
Generic Tree Pruning Procedure

input:
  function f(T, m) (bound/estimate for the generalization error
    of a decision tree T, based on a sample of size m),
  tree T.
foreach node j in a bottom-up walk on T (from leaves to root):
  find T' which minimizes f(T', m), where T' is any of the following:
    the current tree after replacing node j with a leaf 1.
    the current tree after replacing node j with a leaf 0.
    the current tree after replacing node j with its left subtree.
    the current tree after replacing node j with its right subtree.
    the current tree.
  let T := T'.
```

Solution: Overfitting

1. Effect of reduced error pruning



Solution: Overfitting

3. Random forests
   - A classifier consisting of a collection of decision trees (an ensemble of trees)

The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. Given a training set $X = x_1, \ldots, x_n$ with responses $Y = y_1, \ldots, y_n$, bagging repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these samples:

For $b = 1, \ldots, B$:

1. Sample, with replacement, n training examples from X, Y; call these X_b, Y_b.
2. Train a classification or regression tree f_b on X_b, Y_b.

After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x':

$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b(x')$$

or by taking the majority vote in the case of classification trees.

https://en.wikipedia.org/wiki/Random_forest

Real-case example

Practical Applications:

Flight simulator: 20 state variables; 90K examples based on expert pilot's actions; auto-pilot tree
Yahoo Ranking Challenge
Random Forests: Microsoft Kinect Pose Estimation

Real-case example

เทคโนโลยี (Machine Learning) ที่ใช้ใน Microsoft Kinect

https://www.microsoft.com/.../uploads/2016/02/BodyPartRecogni...

Figure from: Raquel Urtasun

- Decision trees are in XBox