



## CS-F415: Data Mining

# 03

## Classification KNN Decision Tree



**Dr. Kamlesh Tiwari**  
Associate Professor, Department of CSIS,  
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA

Feb 02, 2024

MW/F 4:00pm 6101 @ BITS-Pilani [Jan-May 2024]

<http://ktiwarei.in/dm>

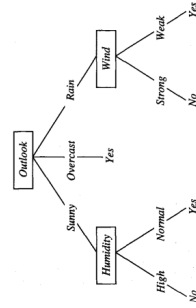
### How to deal with multiple classes

- **Multi-class:** There is more than two labels available for the item to be associated with one
  - 1 One vs All: train multiple classifiers, one for each class, learning whether item belongs to that class or not.
  - 2 One vs One (all pairs): train classifier for every possible pair of classes  $c * (c - 1) / 2$ . Classification is obtained by majority voting
  - 3 Error correction codes
  - 4 Hierarchical methods
  - 5 One-class classification
- **Multi-label:** An item having more than one label

### Decision Tree

#### Decision Tree

A method to approximate discrete-valued functions. It is **robust to noisy** data and capable of learning **disjunctive** expressions. Primarily useful for **classification**.



- Each node in the tree specifies a **test** for some attribute
- Each branch descending from the node corresponds to one of the **possible value**
- Decision trees represent a **disjunction of conjunctions**

$$\vee (\text{Outlook} = \text{Overcast}) \vee (\text{Outlook} = \text{Rain} \wedge \text{Wind} = \text{Weak})$$

$$(\text{Outlook} = \text{Sunny} \wedge \text{Humidity} = \text{Normal})$$

### Classification

Finding the right label



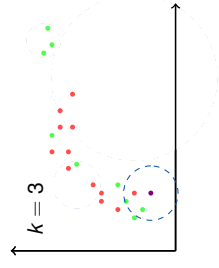
What feature (attributes) would you choose?

Color, texture, weight, density, hardness .....

### K Nearest Neighbor (KNN)

You are most likely as your friends (Bias)

- Two step algorithm
  - 1 Search  $k$  other datum points (most difficult part)
  - 2 Apply majority voting
- A **lazy** learner, **non-parametric** model
- To avoid ties,  $k$  should NOT be a multiple of number of classes
- Small  $k$  is sensitive to noise and large one has high bias



### DT is Appropriate When ...

- Instances are represented by **attribute-value pairs**
- The target function has **discrete** output values
- Disjunctive descriptions may be required
- The training data may contain **errors**
- The training data may contains attributes with **missing value**

## Entropy (Shannon's)

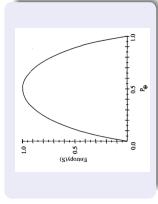
Characterizes the **impurity** of an arbitrary collection of examples

$$Entropy(S) = \sum_{i=1}^c -p_i \log_2 p_i$$

Range is 0 to 1, i.e.  $0 \leq Entropy(S) \leq 1$

**0** – when all members are of same class.

**1** – if equal number of positive and negative



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

$$Entropy([9+, 5-])$$

$$= -(9/14) \log_2(9/14)$$

$$- (5/14) \log_2(5/14)$$

$$= 0.94$$

## Information Gain

$$Gain(S, A) = Entropy(S) - \sum_{v \in Value(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$S_{High} = \{D1, D2, D3, D4, D8, D12, D14\}$$

$$S_{Normal} = \{D5, D6, D7, D9, D10, D11, D13\}$$

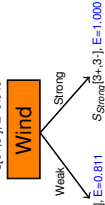
$$S_{High} = \{9+, 5-\}, E=0.940$$

$$S_{Normal} = \{6+, 1-\}, E=0.592$$

$$S_{High} = \{3+, 4-\}, E=0.985$$

$$S_{Weak} = \{9+, 2-\}, E=0.811$$

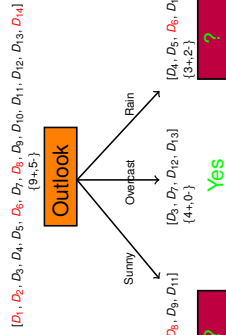
$$S_{Strong} = \{3+, 3-\}, E=1.000$$



$$Gain(S, Humidity) = 0.940 - (7/14)0.985 - (7/14)0.592 = 0.151$$

$$Gain(S, Wind) = 0.940 - (8/14)0.811 - (6/14)1.000 = 0.048$$

## Recursively apply the same



Which attribute to test here?

$$S_{Sunny} = \{D1, D2, D3, D9, D11\}$$

$$Gain(S_{Sunny}, Humidity) = 0.970 - (3/9)0.0 - (2/9)0.0 = 0.970$$

$$Gain(S_{Sunny}, Temperature) = 0.970 - (2/9)0.0 - (2/9)1.0 - (1/9)0.0 = 0.57$$

$$Gain(S_{Sunny}, Wind) = 0.970 - (2/9)1.0 - (3/9)0.0 = 0.019$$

## Information Gain

Information Gain of an attribute,  $A'$  is the expected reduction in entropy caused by partitioning the dataset  $S$  according to that attribute

$$Gain(S, A) = Entropy(S) - \sum_{v \in Value(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

here  $S_v$  is a subset of  $S$  where value of the attribute  $A$  is  $v$

For example

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

And so on....

<sup>1</sup> Outlook, Temperature, Humidity, Wind

## Information Gain and Decision Tree

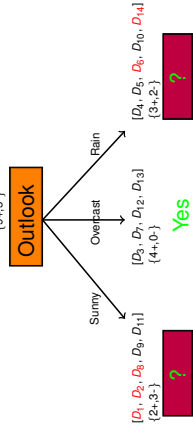
$$Gain(S, Humidity) = 0.151$$

$$Gain(S, Wind) = 0.048$$

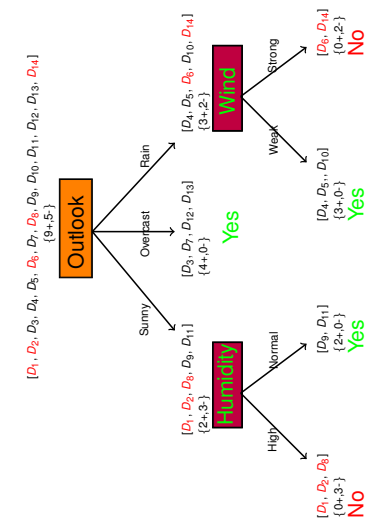
$$Gain(S, Outlook) = 0.246$$

$$Gain(S, Temperature) = 0.029$$

$\{D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12, D13, D14\}$



## Recursively apply the same



## Decision Tree

A method for approximating discrete-valued functions that is robust to noisy data and capable of learning disjunctive expressions

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

What is classification for

(*Outlook* = Rain, *Humidity* = High, *Wind* = Weak)

**ALERT:** (missing value) what is Temperature?

## Iterative-Dichotomiser-3 (ID3) Algorithm By: John Ross Quinlan

**Algorithm 1:** ID3(Examples, Target.attribute, Attributes)

<sup>1</sup> *Examples* are the training data, *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. Algorithm returns a decision tree that correctly classify the given example.

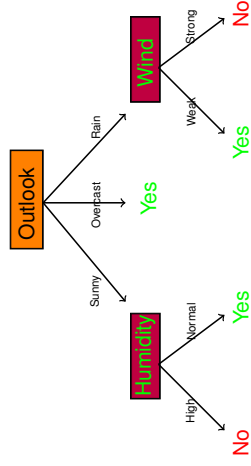
- 2 Create a single-node tree *Root*
- 3 **IF** all *Examples* are +ve **THEN return** *Root* with label +ve
- 4 **IF** all *Examples* are -ve **THEN return** *Root* with label -ve
- 5 **IF** *Attributes* =  $\phi$  **THEN return** *Root* with most common *Target.attribute*
- 6  $A \leftarrow$  attribute from *Attributes* that best classifies *Examples*
- 7 Decision attribute for *Root*  $\leftarrow A$
- 8 **foreach** value  $v_i$  of  $A$  **do**
- 9     Add a new tree branch below *Root*, to test  $A=v_i$
- 10     *Examples* $_{v_i} \leftarrow$  subset of *Examples* having value  $v_i$  for  $A$
- 11     **IF** *Examples* $_{v_i} = \phi$  **THEN** below this branch add a leaf with label = most common value of *Target.attribute* in *Examples*
- 12     **ELSE** below this branch add subtree
- 13     ID3(*Examples* $_{v_i}$ , *Target.attribute*, *Attributes* - { $A$ })

Thank You!

Thank you very much for your attention!

Queries ?

## Example



Classification for (*Outlook* = Rain, *Humidity* = High, *Wind* = Weak) is

YES

## Issues with Decision Tree

Given a collection of training examples, there could be many decision trees could be consistent with the examples

- ID3 search strategy
  - ▶ Selects in favor of shorter trees over longer ones, and
  - ▶ Selects trees that place the attributes with highest information gain closest to the root
- Issues in decision trees include
  - 1 How deeply to grow?
  - 2 Handling continuous attributes
  - 3 Choosing an appropriate attribute selection measure
  - 4 Missing attribute values
  - 5 Attributes with differing costs, and
  - 6 Improving computational efficiency