CO3091 - Computational Intelligence and Software Engineering

Lecture 22

Image from: https://mobileimages.lowes.com/product/converted/034878/034878874647.jpg

Decision Trees — Part II

Leandro L. Minku

Announcements

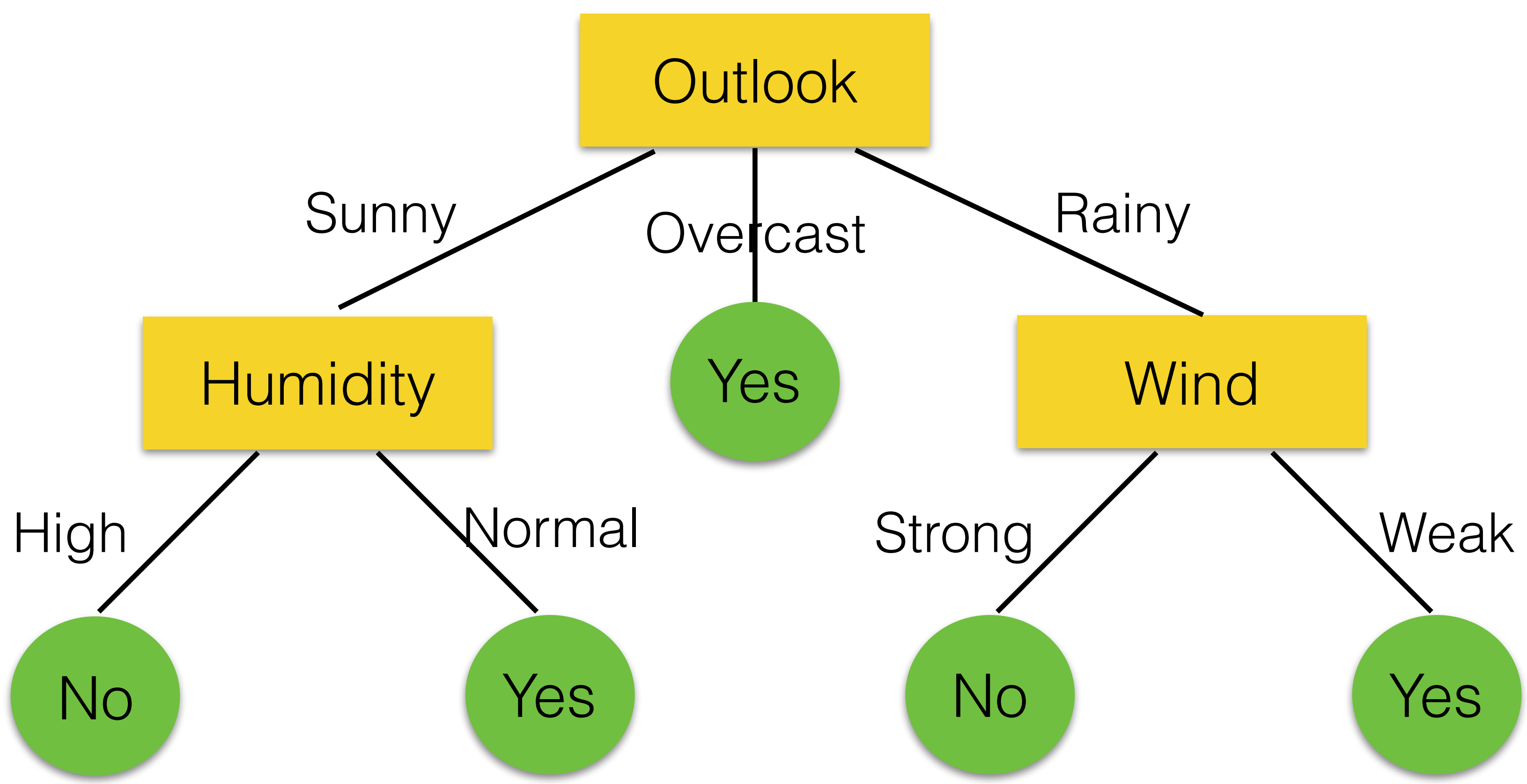
- Results of Coursework 1 are out!
- **Individual** feedback has been sent by email in the report.
  - If you have not received an email, please let me know.
- **General** feedback will be given during the next surgery (tomorrow).
- Marking convention:
  - Each item has been marked separately, except for question 3.
  - The mark for each item is shown in the left of the pages.
  - The overall mark in the coursework is shown in the top of the first page.
  - Blue: items involving implementation and runs, marked by Michael.
  - Red: items involving decisions and analyses, marked by me.
- If you don't understand any comment in the feedback, feel free to contact me.

Overview

- Previous lecture:
  - What are decision trees in the context of machine learning?
  - Recursive algorithm to build decision trees for categorical attributes.
- This lecture:
  - How to determine which attribute is the best one to split on?
- To be continued in the next lecture.
  - How to deal with numerical input attributes?
  - How to deal with overfitting?
  - Applications of decision trees.

Example of Decision Tree

Predictions can be made by walking down the tree.



Irrelevant or redundant attributes don't appear in the tree.

Internal nodes make splits on input attributes.

Leaves indicate output values.

How to Build Decision Trees Based on Training Data?

General idea:

- **Create a node and split it based on the input attribute that best separates the training examples associated to that node into different classes.**

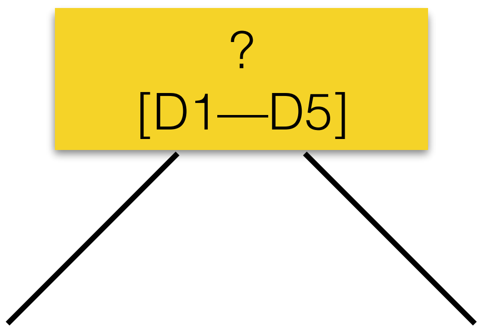
- Once a split is made, create a node for each branch and split it based on the procedure above.

Stopping criteria:

- All training examples associated to the node have the same class.
- There are no more input attributes to split on.
- There are no examples associated to the node.

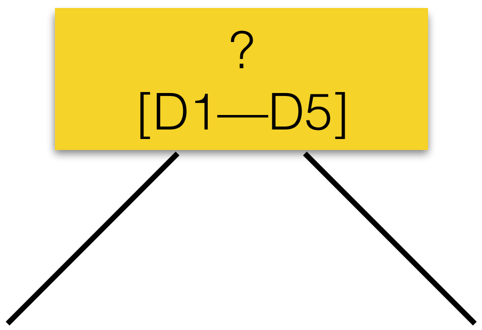
Choosing an Attribute to Split On — Basic Idea

Day	Wind	Humidity	Play
D1	Strong	High	No
D2	Strong	High	No
D3	Weak	High	No
D4	Strong	Normal	Yes
D5	Strong	Normal	Yes



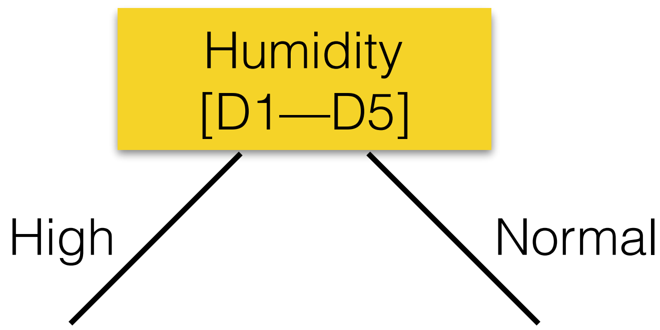
Choosing an Attribute to Split On — Basic Idea

Day	Wind	Humidity	Play
D1	Strong	High	No
D2	Strong	High	No
D3	Weak	High	No
D4	Strong	Normal	Yes
D5	Strong	Normal	Yes



Choosing an Attribute to Split On — Basic Idea

Day	Wind	Humidity	Play
D1	Strong	High	No
D2	Strong	High	No
D3	Weak	High	No
D4	Strong	Normal	Yes
D5	Strong	Normal	Yes

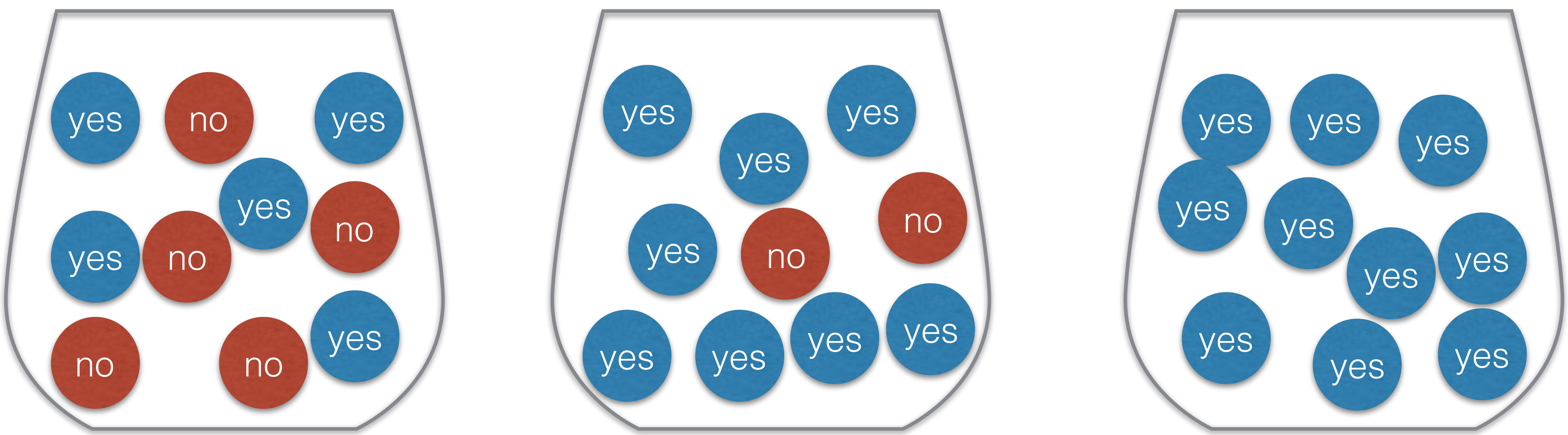


How to compute which attribute best separates the data?

—> for classification problems?

Entropy — General Idea

Entropy characterises the impurity of a collection of examples.



Very impure
= high entropy

Less impure
= smaller entropy

Very pure
= low entropy

Entropy for Two Classes

Given a collection of examples whose class can be yes or no, the entropy of the examples is:

$$\text{Entropy}(\text{examples}) = -\,p_{yes} \log_2(p_{yes}) - p_{no} \log_2(p_{no})$$

where p_{yes} is the proportion of examples from class yes and p_{no} is the proportion of examples from class no.

Entropy when $p_{ci} = 0$

When $p_{ci} = 0$, we say that $-p_{ci} \log_2(p_{ci})$ is zero.

where ci is a given class of your machine learning problem (e.g., yes or no)

Example of Entropy

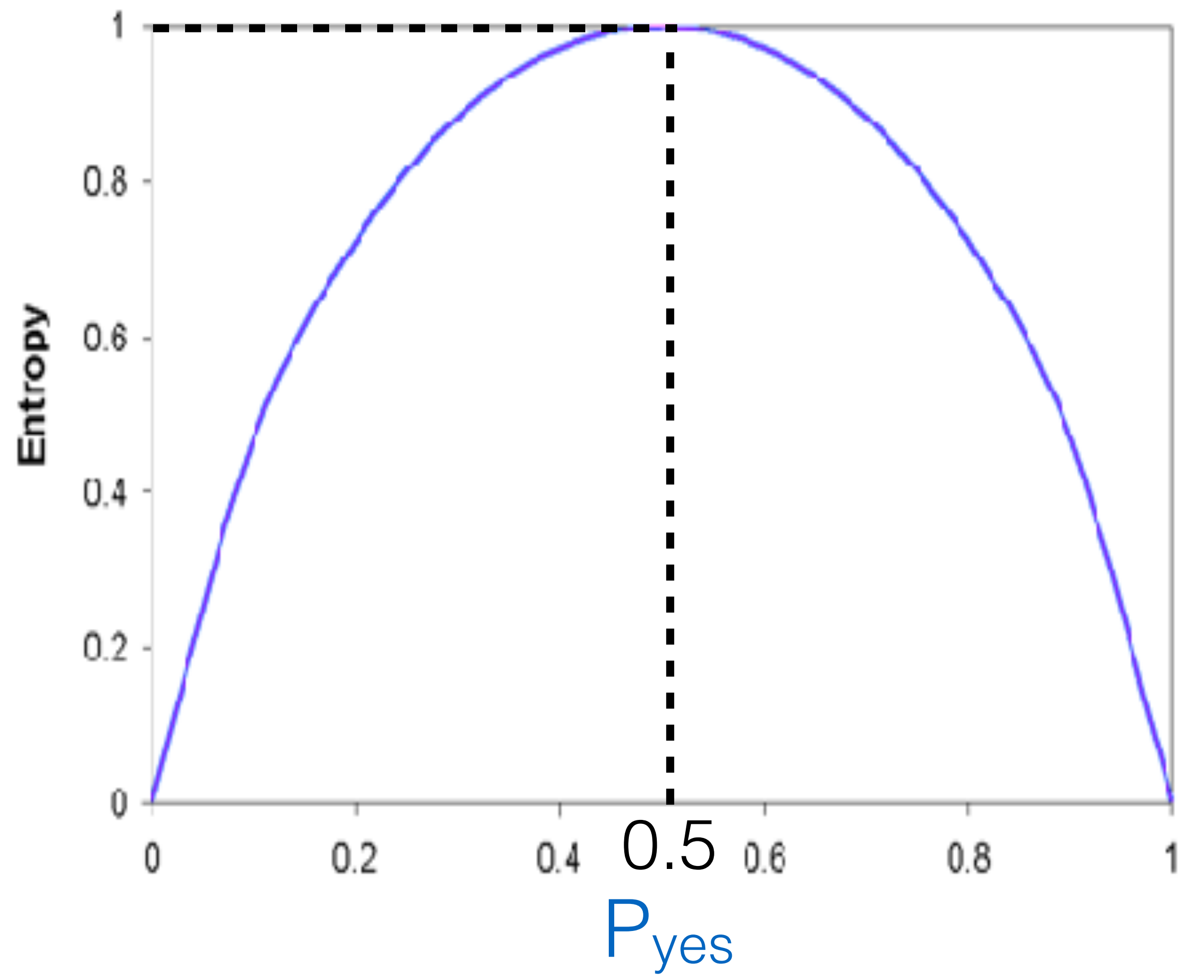
$$\text{Entropy(examples)} = -\, p_{yes} \log_2(p_{yes}) - p_{no} \log_2(p_{no})$$

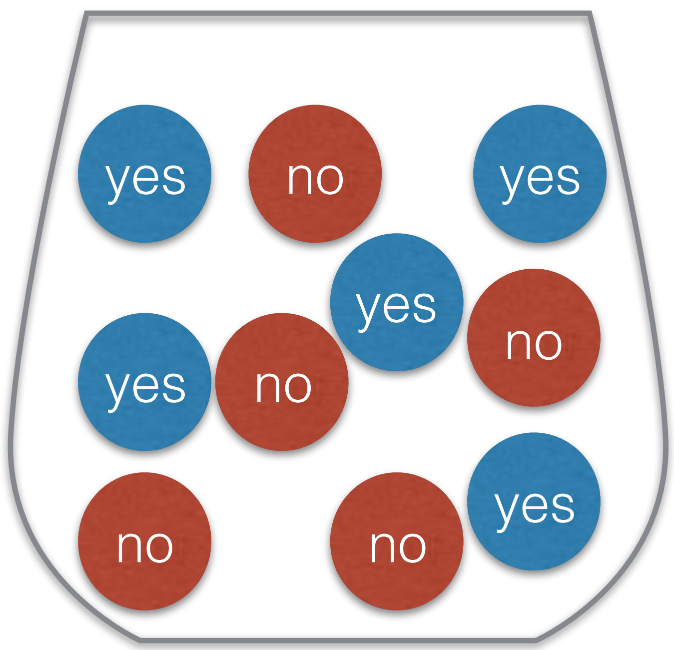
where p_{yes} is the proportion of examples from class yes and p_{no} is the proportion of examples from class no.

Day	Wind	Humidity	Play
D1	Strong	High	No
D2	Strong	High	No
D3	Weak	High	No
D4	Strong	Normal	Yes
D5	Strong	Normal	Yes

$$\text{Entropy(D1–D5)} = -\, 2/5 \ \log_2(2/5) \ -3/5 \ \log_2(3/5) \approx 0.53 + 0.44 = 0.97$$

Entropy for Two Classes

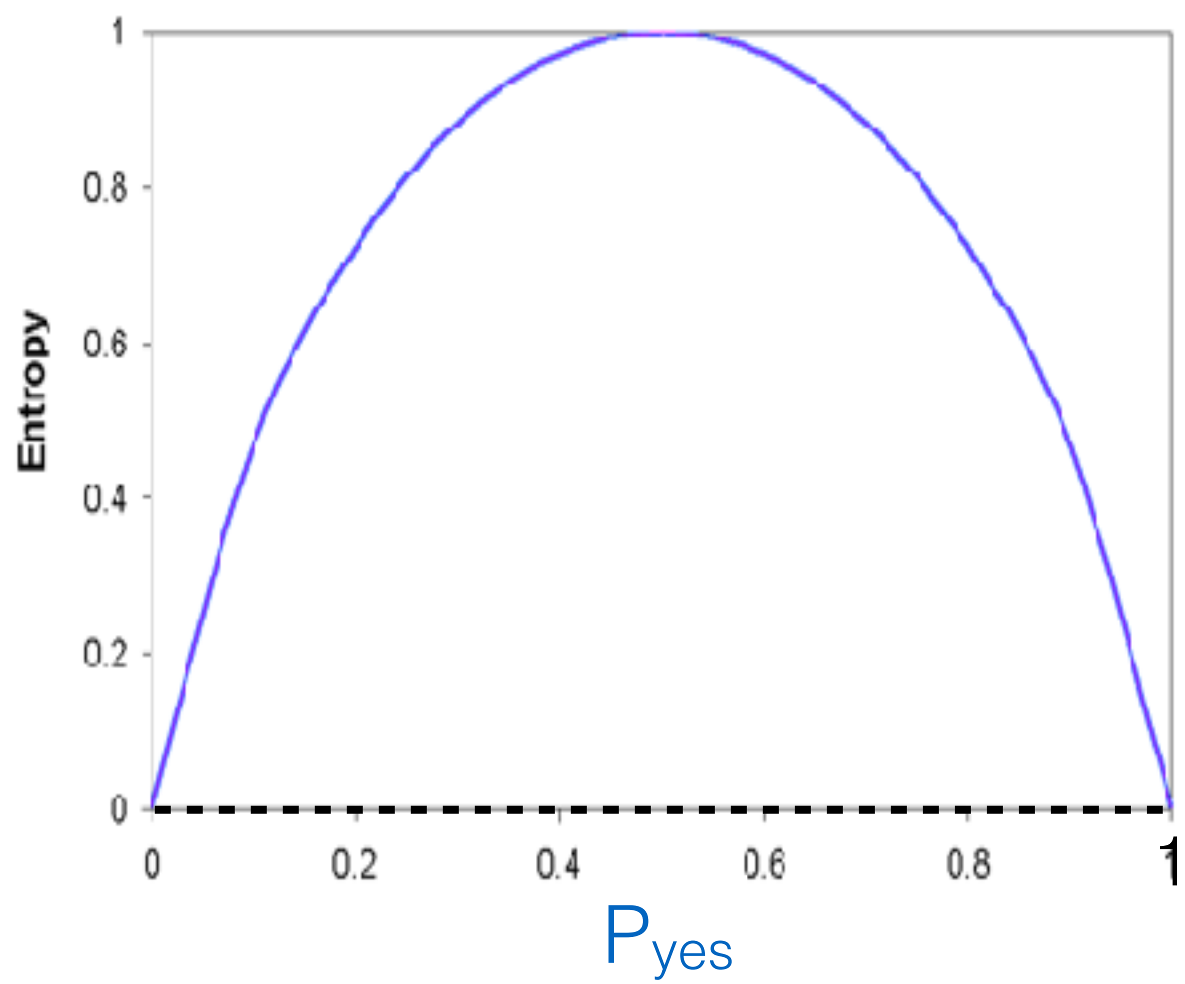


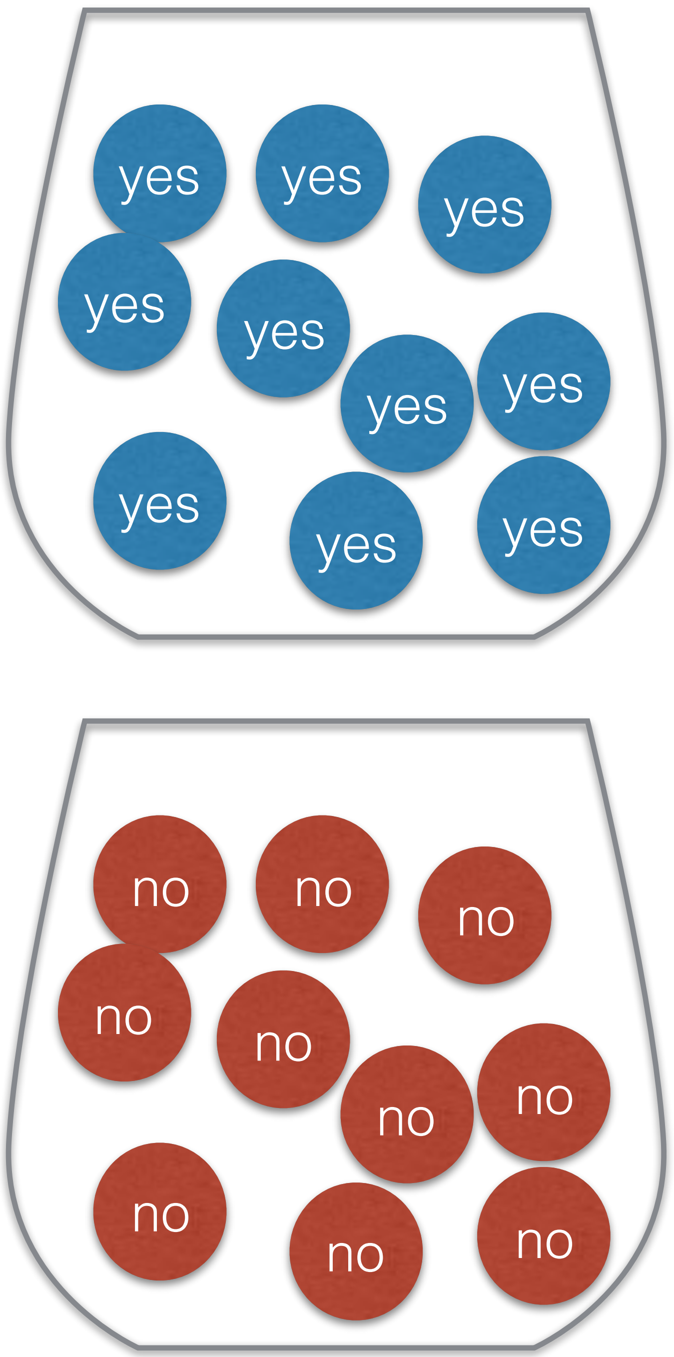


If half of the examples are from class yes and half from class no, entropy has its maximum value — the data are very impure.

Image from: http://dms.irb.hr/tutorial/images/entropy_graph.gif

Entropy for Two Classes





If all examples are from a single class, entropy has its minimum value — the data are not impure.

Image from: http://dms.irb.hr/tutorial/images/entropy_graph.gif

Entropy For k Classes

Given a collection of examples whose classes can be from c_1 to c_k:

$$\text{Entropy(examples)} = - p_{c1} \log_2(p_{c1}) - p_{c2} \log_2(p_{c2}) - \ldots - p_{ck} \log_2(p_{ck})$$

where p_{ci} is the proportion of examples from class c_i, $1 \leq i \leq k$.

Information Gain — General Idea

For classification problems, the input attribute that best separates the data is the one that provides the largest reduction in entropy (information gain).

Day	Wind	Humidity	Play
D1	Strong	High	No
D2	Strong	High	No
D3	Weak	High	No
D4	Strong	Normal	Yes
D5	Strong	Normal	Yes

high entropy (impure)

Day	Wind	Humidity	Play
D1	Strong	High	No
D2	Strong	High	No
D3	Weak	High	No

low entropy (pure)

Day	Wind	Humidity	Play
D4	Strong	Normal	Yes
D5	Strong	Normal	Yes

Splitting on humidity is "very informative" (high information gain).

Information Gain — General Idea

The input attribute that best separates the data is the one that provides the largest reduction in entropy (information gain).

Day	Wind	Humidity	Play
D1	Strong	High	No
D2	Strong	High	No
D3	Weak	High	No
D4	Strong	Normal	Yes
D5	Strong	Normal	Yes

high entropy (impure)

Day	Wind	Humidity	Play
D3	Weak	High	No

low entropy (pure)

Day	Wind	Humidity	Play
D1	Strong	High	No
D2	Strong	High	No
D4	Strong	Normal	Yes
D5	Strong	Normal	Yes

high entropy (impure)

Splitting on wind is "less informative" (less information gain).

Information Gain

- Information gain: measures the reduction in entropy (impurity) when we split the data based on a given attribute.
- Given a collection of examples and an attribute A, the information gain of splitting examples using A is:

Current entropy

Overall entropy of subsets after split

Entropy of subset of examples with A = vi

$$\text{InfoGain}(\text{examples}, A) = \text{Entropy}(\text{examples}) - \sum_{v_i \in \text{Values}(A)} \frac{|\text{examples}_{v_i}|}{|\text{examples}|} \text{Entropy}(\text{examples}_{v_i})$$

Where:
Values(A) are all possible values that attribute A can assume,
examples_{v_i} is the collection of examples whose attribute A has value v_i.

Example of Information Gain

Day	Wind	Humidity	Play
D1	Strong	High	No
D2	Strong	High	No
D3	Weak	High	No
D4	Strong	Normal	Yes
D5	Strong	Normal	Yes

$$\text{InfoGain(examples,A)} = \text{Entropy(examples)} - \sum_{v_i \in \text{Values(A)}} \frac{|\text{examples}_{vi}|}{|\text{examples}|} \text{Entropy}(\text{examples}_{vi})$$

$$\text{InfoGain(D1-D5,Wind)} = 0.97 - \underbrace{\boxed{\frac{4}{5} \times 1}}_{v_i\,=\,\text{Strong}} - \underbrace{\boxed{\frac{1}{5} \times 0}}_{v_i\,=\,\text{Weak}} = 0.97 - 0.80 = 0.17$$

Example of Information Gain

Day	Wind	Humidity	Play
D1	Strong	High	No
D2	Strong	High	No
D3	Weak	High	No
D4	Strong	Normal	Yes
D5	Strong	Normal	Yes

$$\text{InfoGain(examples,A)} = \text{Entropy(examples)} - \sum_{v_i \in \text{Values(A)}} \frac{|\text{examples}_{vi}|}{|\text{examples}|} \text{Entropy}(\text{examples}_{vi})$$

$$\text{InfoGain(D1-D5,Humidity)} = 0.97 - \underbrace{\boxed{\frac{3}{5} \times 0}}_{v_i = \text{High}} - \underbrace{\boxed{\frac{2}{5} \times 0}}_{v_i = \text{Normal}} = 0.97 - 0 = 0.97$$

Example of Information Gain

Day	Wind	Humidity	Play
D1	Strong	High	No
D2	Strong	High	No
D3	Weak	High	No
D4	Strong	Normal	Yes
D5	Strong	Normal	Yes

$$\text{InfoGain}(\text{D1-D5}, \text{Wind}) = 0.97 - \underbrace{\boxed{\frac{4}{5} \times 1}}_{v_i = \text{Strong}} - \underbrace{\boxed{\frac{1}{5} \times 0}}_{v_i = \text{Weak}} = 0.97 - 0.80 = 0.17$$

Split on humidity provides more information gain.

$$\text{InfoGain}(\text{D1-D5}, \text{Humidity}) = 0.97 - \underbrace{\boxed{\frac{3}{5} \times 0}}_{v_i = \text{High}} - \underbrace{\boxed{\frac{2}{5} \times 0}}_{v_i = \text{Normal}} = 0.97 - 0 = 0.97$$

Choosing Input Attribute to Split in Classification Problems

- Given a set of candidate input attributes, make the split on the attribute that leads to the highest information gain.
  - This is the attribute that is going to reduce the entropy (impurity) the most.
  - This is the attribute that is going to best separate the examples into different classes.

How to compute which attribute best separates the data?

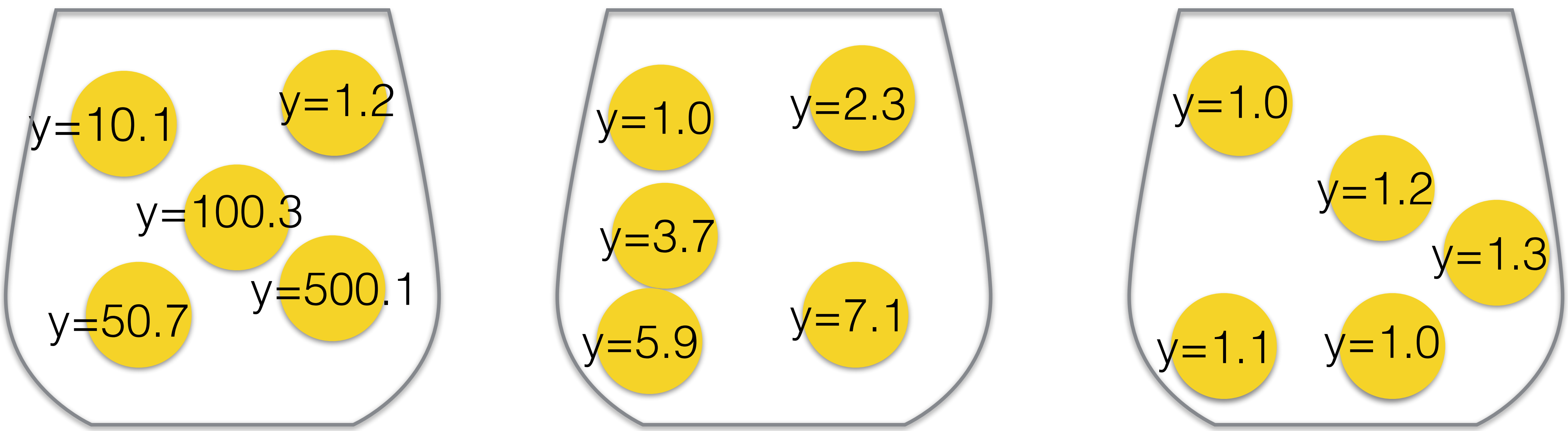
—> for regression problems?

Splitting Nodes in Regression Trees — Categorical or Ordinal Input Attributes

- In classification trees, we make splits based on information gain.
- Information gain = reduction in entropy.
- Entropy is a measure of impurity for classification problems.

How to measure "impurity" for regression problems?

Reduction in Variance



Very heterogeneous = high variance ≈ high entropy

Less heterogeneous = smaller variance ≈ smaller entropy

Very homogeneous = low variance ≈ low entropy

Variance of Output Values

Given a collection of examples, the variance of their **output** values is:

$$\text{Variance}(\text{examples}) = \frac{1}{|\text{examples}|} \sum_{(\mathbf{x}_i, y_i) \in \text{examples}} [y_i - \text{mean}(\text{examples})]^2$$

Where:

y_i is the output value of example i.
mean(examples) is the mean of all **output** values y_i , $(\mathbf{x}_i, y_i) \in$ examples.

Variance of Output Values

Project	Size	Team Expertise	Effort
P1	Small	High	1
P2	Small	High	2
P3	Medium	High	3
P4	Medium	Normal	4
P5	Large	Normal	10

$$\text{Variance}(\text{examples}) = \frac{1}{|\text{examples}|} \sum_{(\mathbf{x}_i, y_i) \in \text{examples}} [y_i - \text{mean}(\text{examples})]^2$$

$$\text{Variance}(\text{P1–P5}) = \frac{1}{5} \times \left([1 - 4]^2 + [2 - 4]^2 + [3 - 4]^2 + [4 - 4]^2 + [10 - 4]^2\right)$$

$$= \frac{1}{5} \times (9 + 4 + 1 + 0 + 36) = \frac{50}{5} = 10$$

Reduction in Variance

$$\text{InfoGain}(\text{examples},A) = \text{Entropy}(\text{examples}) - \sum_{v_i \in \text{Values}(A)} \frac{|\text{examples}_{vi}|}{|\text{examples}|} \text{Entropy}(\text{examples}_{vi})$$

$$\text{VarRed}(\text{examples},A) = \text{Variance}(\text{examples}) - \sum_{v_i \in \text{Values}(A)} \frac{|\text{examples}_{vi}|}{|\text{examples}|} \text{Variance}(\text{examples}_{vi})$$

Choosing Input Attribute to Split in Regression Problems

- Given a set of candidate input attributes, make the split on the attribute that leads to the highest reduction in variance.
  - This is the attribute that is going to reduce the heterogeneity (variance) the most.
  - This is the attribute that is going to best separate the examples into different sets.

How to Build Decision Trees Based on Training Data?

General idea:

- Create a node and split it based on the input attribute that best separates the training examples associated to that node into different classes.

- Once a split is made, create a node for each branch and split it based on the procedure above.

Stopping criteria:

- All training examples associated to the node have the same class.
- There are no more input attributes to split on.
- There are no examples associated to the node.

How to Build Decision Trees Based on Training Data?

General idea:

- Create a node and split it based on the input attribute that best separates the training examples associated to that node into different classes —> based on InfoGain or VarRed.

- Once a split is made, create a node for each branch and split it based on the procedure above.

Stopping criteria:

- All training examples associated to the node have the same class.
- There are no more input attributes to split on.
- There are no examples associated to the node.

DecisionTreeLearning (**examples, input_attributes**, output_attribute)

1. Create a **root** node for the tree, associated to the **examples**

2. If all **examples** belong to the same class (or numerical value), return the **root** node as leaf node of that class.

3. If **input_attributes** is empty, return the **root** node as leaf node of the majority class (or average) among the **examples**.

4. **A** <— attribute in **input_attributes** that leads to the highest information gain (or reduction in variance)

5. For each possible value $\mathbf{v_i}$ of **A**

    5.1 Add a new tree branch below **root** corresponding to **A** = $\mathbf{v_i}$

    5.2 Let $\mathbf{examples_{vi}}$ be the subset of **examples** with **A** = $\mathbf{v_i}$

    5.3 If $\mathbf{examples_{vi}}$ is empty

        5.3.1 Add a leaf node below this branch using the majority class (or average) among **examples**

    5.4 Else

        5.4.1 Add the following subtree below this branch:

        DecisionTreeLearning($\mathbf{examples_{vi}}$, **input_attributes \ {A}**, output_attribute)

6. Return root

Summary

- Choosing the best attribute for a split:
  - Classification problems: highest information gain.
  - Regression problems: highest reduction in variance.
- Pseudocode for decision trees.
- Next surgery:
  - Decision tree exercises — bring your calculators.
- Next lecture:
  - How to deal with numerical input attributes?
  - How to deal with overfitting?
  - Applications of decision trees.

Further Reading

Tom Mitchell

Machine Learning

London : McGraw-Hill, 1997

Chapter 3, sections 3.1 to 3.5.

http://www.cs.princeton.edu/courses/archive/spr07/cos424/papers/mitchell-dectrees.pdf

Menzies et al.

Sharing Data and Models in Software Engineering

Elsevier, 2014

Section 10.10 (Extensions for Continuous Classes)

http://www.sciencedirect.com/science/book/9780124172951