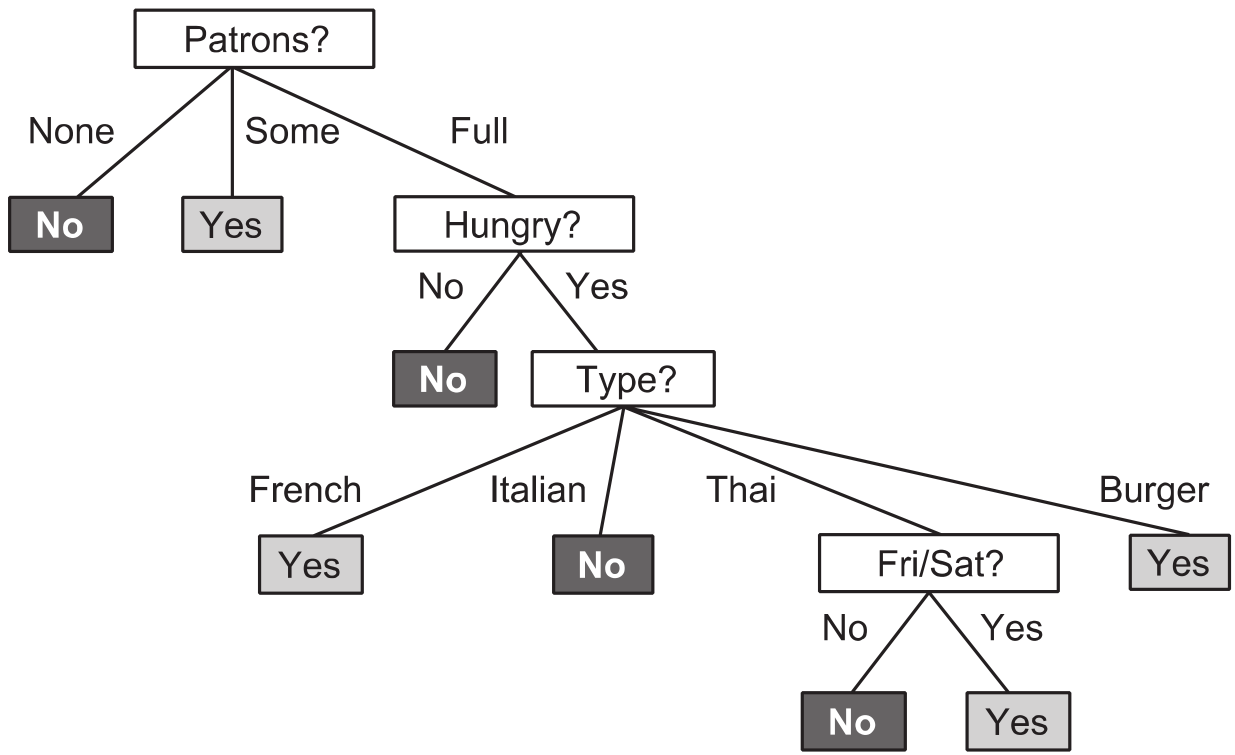
Lecture 2: Decision Tree

Course Teacher: Md. Shariful Islam Bhuyan

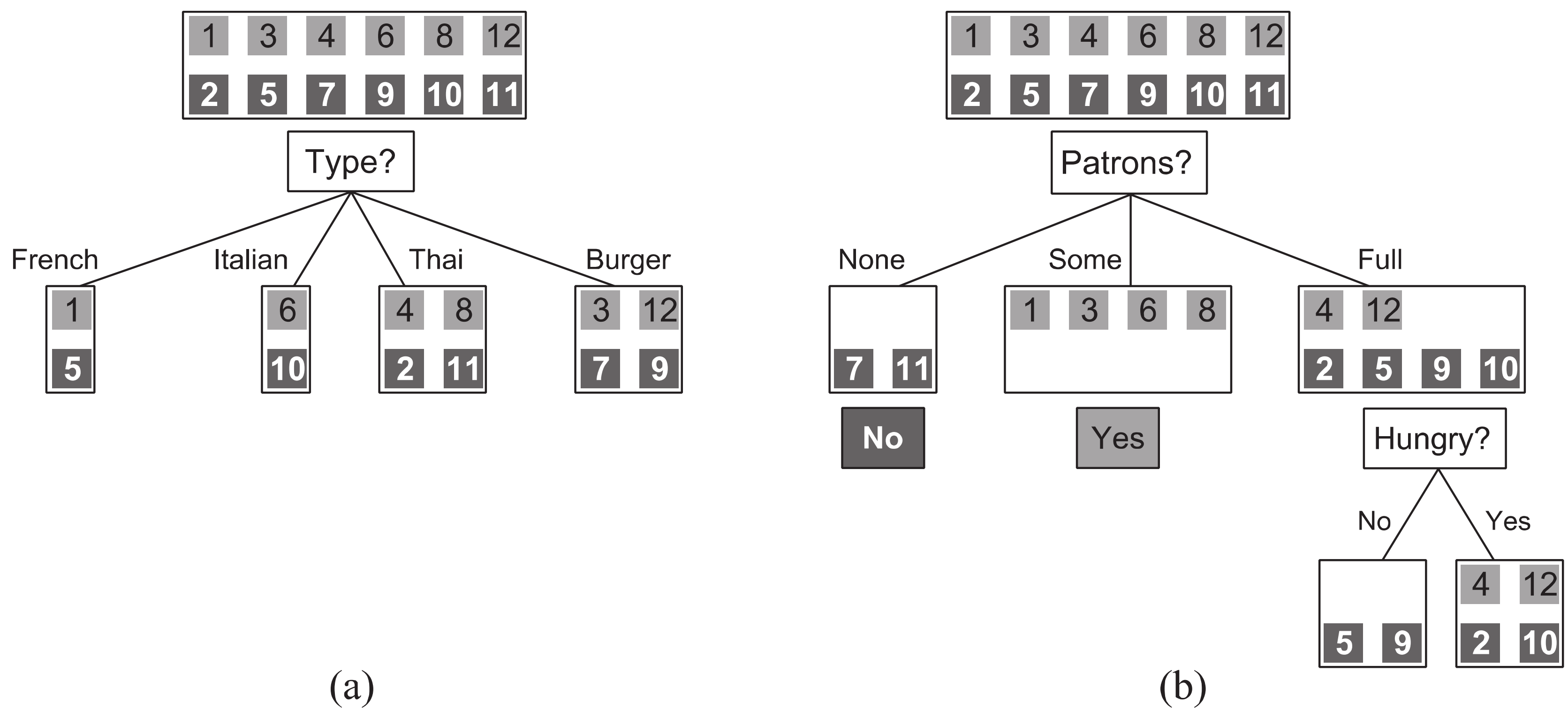
Example: samples

Sample	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Goal
x_1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0–10	y_1 = Yes
x_2	Yes	No	No	Yes	Full	$	No	No	Thai	30–60	y_2 = No
x_3	No	Yes	No	No	Some	$	No	No	Burger	0–10	y_3 = Yes
x_4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10–30	y_4 = Yes
x_5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	y_5 = No
x_6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0–10	y_6 = Yes
x_7	No	Yes	No	No	None	$	Yes	No	Burger	0–10	y_7 = No
x_8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0–10	y_8 = Yes
x_9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	y_9 = No
x_{10}	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10–30	y_{10} = No
x_{11}	No	No	No	No	None	$	No	No	Thai	0–10	y_{11} = No
x_{12}	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30–60	y_{12} = Yes

Decision tree



Which is better split?



Better Attribute

- Divides example into sets with
  - less uniform distribution or less entropy
- Entropy is a measure of the uncertainty of a random variable
  - For binary variable $B(q) = -q \log_2 q - (1-q)\log_2(1-q)$
- Information gain $G(A) = B\left(\frac{p}{p+n}\right) - \sum_{k=1}^{d} \frac{p_k}{p_k+n_k} B\left(\frac{p}{p+n}\right)$

Inducing decision trees from examples

- Four cases for subproblems
  1. All positive (or all negative): we are done
  2. Some positive and some negative: choose next **important** attribute (greedy)
  3. No examples left: unobserved case, use prior knowledge/plurality
  4. No attributes left: error, noise, partial information, inherent uncertainty

Decision tree pseudocode

```
function DECISION-TREE-LEARNING(examples, attributes, parent examples) returns a tree
    if examples is empty then return PLURALITY-VALUE(parent examples)
    else if all examples have the same classification then return the classification
    else if attributes is empty then return PLURALITY-VALUE(examples)
    else
        A ← argmax_{a ∈ attributes} IMPORTANCE(a, examples)
        tree ← a new decision tree with root test A
        for each value v_k of A do
                exs ← { e : e ∈ examples and e.A = v_k }
                subtree ← DECISION-TREE-LEARNING(exs, attributes – A, examples)
                add a branch to tree with label (A = v_k) and subtree subtree
        return tree
```

Assignment 1

- In Python
  - https://www.quora.com/How-should-I-learn-Python-for-machine-learning-and-artificial-intelligence
- Decision tree implementation
- Adaboost implementation
- Performance measure: Accuracy, Precision, Recall, F1-score
  - https://en.wikipedia.org/wiki/Confusion matrix
- Detect overfitting: Training-test split

Overfitting

- Generate a large tree when there is actually no pattern to be found
  - Example: roll of a die will come up as 6 or not.
- Irrelevant attribute: color, weight, time, is fingers crossed
- For fair dice, learn a tree with a single node that says "NO"
- Overfitting: 2 rolls of a 7-gram blue die with fingers crossed
- More likely as the number of input attributes grows
- Less likely as we increase the number of training examples.