

Lecture 3: 16 January, 2024

Madhavan Mukund

<https://www.cmi.ac.in/~madhavan>

Data Mining and Machine Learning
January–April 2024

Supervised learning

- A set of items
 - Each item is characterized by attributes (a_1, a_2, \dots, a_k)
 - Each item is assigned a class or category c
- Given a set of examples, predict c for a new item with attributes $(a'_1, a'_2, \dots, a'_k)$
- Examples provided are called **training data**
- Aim is to **learn** a mathematical model that **generalizes** the training data
 - Model built from training data should extend to previously unseen inputs
- **Classification** problem
 - Usually assumed to binary — two classes

Example: Loan application data set

| ID | Age | Has_job | Own_house | Credit_rating | Class |
|----|--------|---------|-----------|---------------|------------|
| 1 | young | false | false | fair | No |
| 2 | young | false | false | good | No |
| 3 | young | true | false | good | Yes |
| 4 | young | true | true | fair | Yes |
| 5 | young | false | false | fair | No |
| 6 | middle | false | false | fair | No |
| 7 | middle | false | false | good | No |
| 8 | middle | true | true | good | Yes |
| 9 | middle | false | true | excellent | Yes |
| 10 | middle | false | true | excellent | Yes |
| 11 | old | false | true | excellent | Yes |
| 12 | old | false | true | good | Yes |
| 13 | old | true | false | good | Yes |
| 14 | old | true | false | excellent | Yes |
| 15 | old | false | false | fair | No |

Basic assumptions

Fundamental assumption of machine learning

- Distribution of training examples is identical to distribution of unseen data

What does it mean to learn from the data?

- Build a model that does better than random guessing
 - In the loan data set, always saying **Yes** would be correct about **9/15** of the time
- Performance should ideally improve with more training data

How do we evaluate the performance of a model?

- Model is optimized for the training data. How well does it work for unseen data?
- Don't know the correct answers in advance to compare — different from normal software verification

The road ahead

Many different models

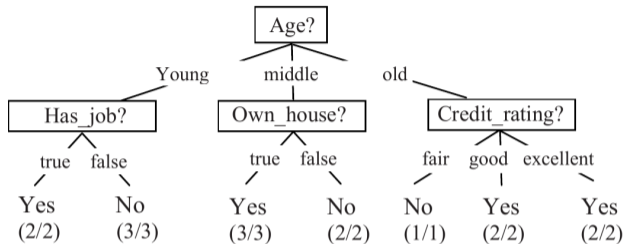
- Decision trees
- Probabilistic models — naïve Bayes classifiers
- Models based on geometric separators
 - Support vector machines (SVM)
 - Neural networks

Important issues related to supervised learning

- Evaluating models
- Ensuring that models generalize well to unseen data
 - A theoretical framework to provide some guarantees
- Strategies to deal with the training data bottleneck

Decision trees

- Play “20 Questions” with the training data
- Query an attribute
 - Partition the training data based on the answer
- Repeat until you reach a partition with a uniform category
- Queries are **adaptive**
 - Different along each path, depends on history



| ID | Age | Has_job | Own_house | Credit_rating | Class |
|----|--------|---------|-----------|---------------|-------|
| 1 | young | false | false | fair | No |
| 2 | young | false | false | good | No |
| 3 | young | true | false | good | Yes |
| 4 | young | true | true | fair | Yes |
| 5 | young | false | false | fair | No |
| 6 | middle | false | false | fair | No |
| 7 | middle | false | false | good | No |
| 8 | middle | true | true | good | Yes |
| 9 | middle | false | true | excellent | Yes |
| 10 | middle | false | true | excellent | Yes |
| 11 | old | false | true | excellent | Yes |
| 12 | old | false | true | good | Yes |
| 13 | old | true | false | good | Yes |
| 14 | old | true | false | excellent | Yes |
| 15 | old | false | false | fair | No |

Decision tree algorithm

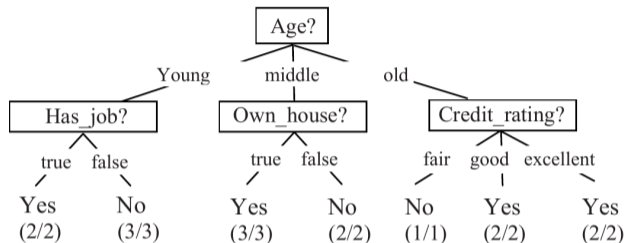
A : current set of attributes

Pick $a \in A$, create children corresponding to resulting partition with attributes $A \setminus \{a\}$

Stopping criterion:

- Current node has uniform class label
- A is empty — no more attributes to query

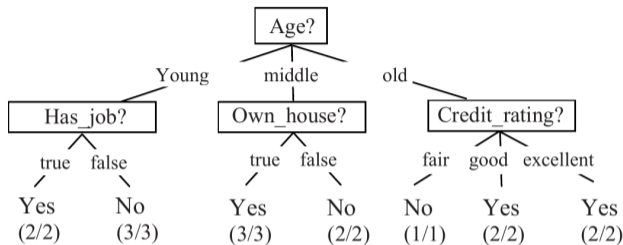
If a leaf node is not uniform, use majority class as prediction



- Non-uniform leaf node — identical combination of attributes, but different classes
- Attributes do not capture all criteria used for classification

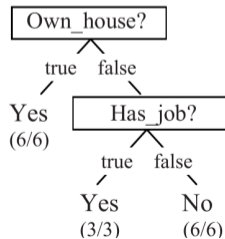
Decision trees

- Tree is not unique
- Which tree is better?
- Prefer small trees
 - Explainability
 - Generalize better (see later)



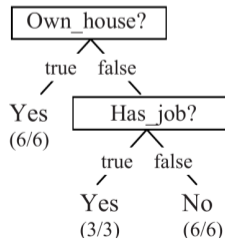
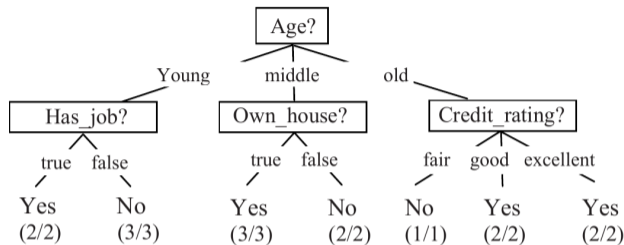
Unfortunately

- Finding smallest tree is NP-complete — for any definition of “smallest”
- Instead, greedy heuristic



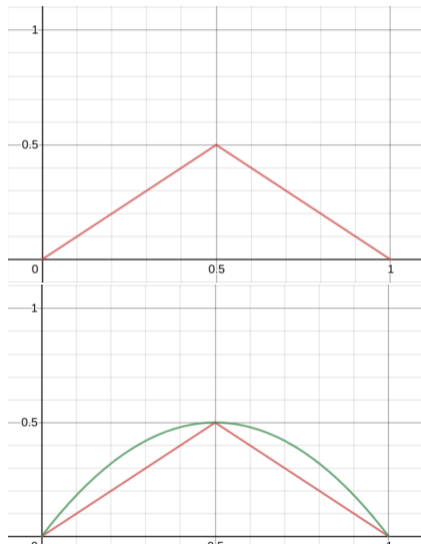
Greedy heuristic

- Goal: partition with uniform category — **pure** leaf
- Impure node — best prediction is majority value
- Minority ratio is **impurity**
- Heuristic: reduce impurity as much as possible
- For each attribute, compute weighted average impurity of children
- Choose the minimum



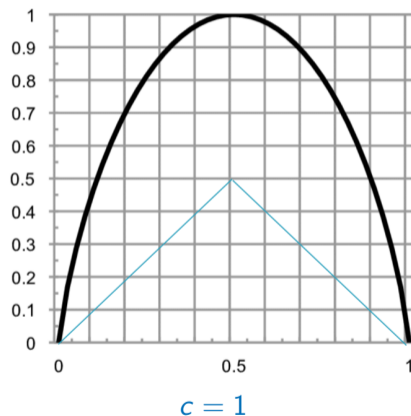
A better impurity function

- Misclassification rate is linear
- Impurity measure that increases more sharply performs better, empirically
- Entropy — [Quinlan]
- Gini index — [Breiman]



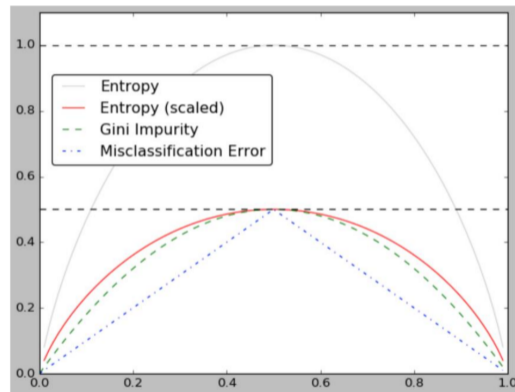
Entropy

- Information theoretic measure of randomness
- Minimum number of bits to transmit a message — [Shannon]
- n data items
 - n_0 with $c = 0$, $p_0 = n_0/n$
 - n_1 with $c = 1$, $p_1 = n_1/n$
- Entropy
$$E = -(p_0 \log_2 p_0 + p_1 \log_2 p_1)$$
- Minimum when $p_0 = 1, p_1 = 0$ or vice versa — note, declare $0 \log_2 0$ to be 0
- Maximum when $p_0 = p_1 = 0.5$



Gini Index

- Measure of unequal distribution of wealth
- Economics — [Corrado Gini]
- As before, n data items
 - n_0 with $c = 0$, $p_0 = n_0/n$
 - n_1 with $c = 1$, $p_1 = n_1/n$
- **Gini Index** $G = 1 - (p_0^2 + p_1^2)$
- $G = 0$ when $p_0 = 0$, $p_1 = 0$ or v.v.
 $G = 0.5$ when $p_0 = p_1 = 0.5$
- Entropy curve is slightly steeper, but Gini index is easier to compute
- Decision tree libraries usually use Gini index



$c = 1$