Pattern Recognition

Lecture 19. Decision Trees

Dr. Shanshan ZHAO

School of AI and Advanced Computing
Xi'an Jiaotong-Liverpool University

Academic Year 2023-2024

Outline

1. Introduction
2. Key Concepts
3. How Decision Trees Work
4. Decision Tree Construction
5. Order of Testing Features

1. Introduction
2. Key Concepts
3. How Decision Trees Work
4. Decision Tree Construction
5. Order of Testing Features

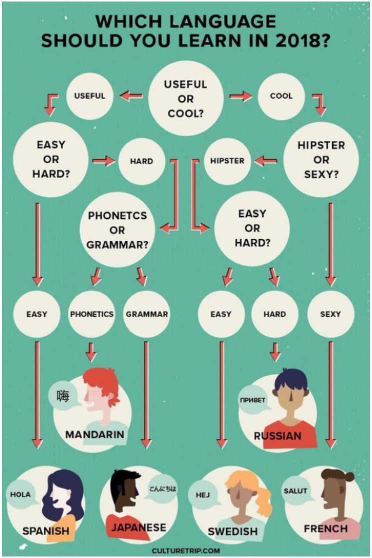
- feature vectors of real-valued and discrete-valued numbers
  - in all cases there has been a natural measure of distance between such vectors
- nominal data
  - discrete and without any natural notion of similarity or even ordering
  - how can we efficiently learn categories using such non-metric data?
    - rule-based or syntactic pattern recognition methods

What is a Decision Tree?

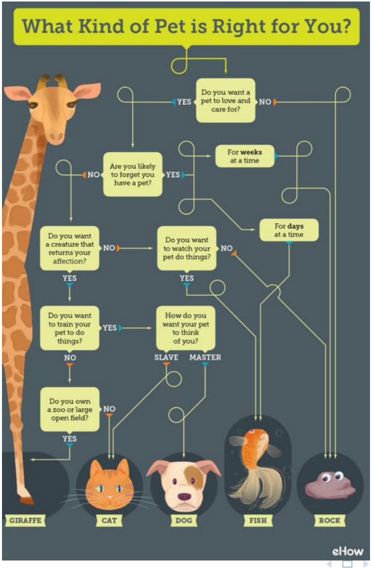
A decision tree is a popular and widely used machine learning algorithm that is primarily employed for both classification and regression tasks.

- It is a visual representation of a decision-making process and can be thought of as a flowchart-like structure that consists of nodes and branches.
- Decision trees are often used in data analysis and predictive modeling to make decisions or predictions based on a set of conditions or rules.

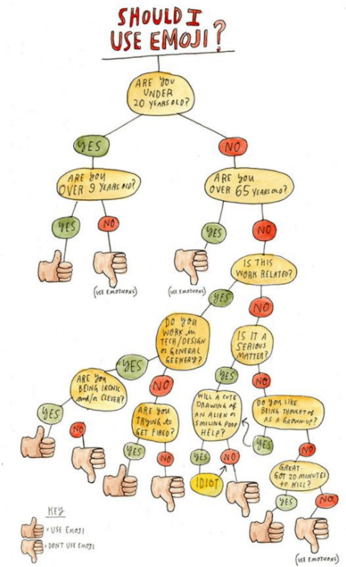
Example: Which language should you learn?



Example: What kind of pet is right for you?


What Kind of Pet is Right for You?
Do you want a pet to love and care for?
YES
NO
Are you likely to forget you have a pet?
NO
YES
For weeks at a time
For days at a time
Do you want a creature that returns your affection?
NO
Do you want to watch your pet do things?
NO
YES
YES
Do you want to train your pet to do things?
YES
How do you want your pet to think of you?
SLAVE
MASTER
NO
Do you own a zoo or large open field?
NO
YES
GIRAFFE
CAT
DOG
FISH
ROCK
eHow

Example: Should you use emoji in a conversation?


SHOULD I USE EMOJI?
ARE YOU UNDER 20 YEARS OLD?
YES
NO
ARE YOU OVER 9 YEARS OLD?
ARE YOU OVER 65 YEARS OLD?
YES
NO
(USE EMOTICONS)
YES
NO
(USE EMOTICONS)
IS THIS WORK RELATED?
YES
NO
DO YOU WORK IN TECH/DESIGN OR GENERAL GEEKERY?
IS IT A SERIOUS MATTER?
YES
NO
ARE YOU BEING IRONIC AND/OR CLEVER?
YES
NO
ARE YOU TRYING TO GET FIRED?
YES
NO
YES
NO
WILL A CUTE DRAWING OF AN ALIEN OR SMILING POOP HELP?
DO YOU LIKE BEING THOUGHT OF AS A GROWN-UP?
YES
NO
IDIOT
YES
NO
GREAT. GOT 20 MINUTES TO KILL?
YES
NO
(USE EMOTICONS)
KEY
= USE EMOJI
= DON'T USE EMOJI

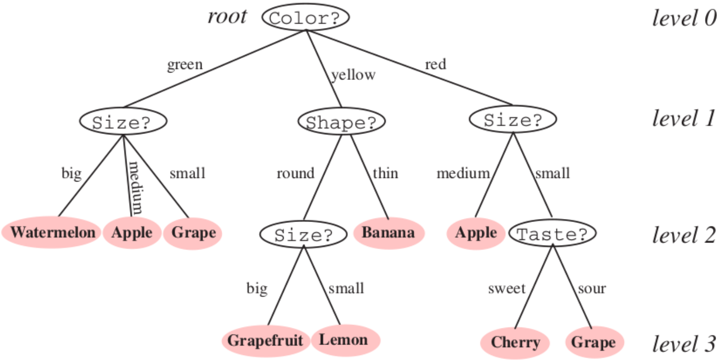


Figure 8.1: Classification in a basic decision tree proceeds from top to bottom. The questions asked at each node concern a particular property of the pattern, and the downward links correspond to the possible values. Successive nodes are visited until a terminal or leaf node is reached, where the category label is read. Note that the same question, `Size?`, appears in different places in the tree, and that different questions can have different numbers of branches. Moreover, different leaf nodes, shown in pink, can be labeled by the same category (e.g., **Apple**).

1 Introduction

2 **Key Concepts**

3 How Decision Trees Work

4 Decision Tree Construction

5 Order of Testing Features

Key Concepts and Terminology

- Nodes: A decision tree is composed of nodes. There are two main types of nodes:
  - Root Node: The topmost node that represents the initial decision or the starting point of the tree.
  - Internal Nodes: Nodes within the tree that represent decisions or conditions. They lead to other nodes in the tree.
- Leaves (Terminal Nodes): Terminal nodes, or leaves, represent the final outcomes or decisions. In classification problems, leaves correspond to class labels, while in regression problems, they represent predicted values.
- Edges (Branches): Edges connect nodes and represent the decision path from one node to another.
- Attributes (Features): Decision trees make decisions based on the values of attributes or features. These attributes are used to split the data at internal nodes.

1. Introduction
2. Key Concepts
3. How Decision Trees Work
4. Decision Tree Construction
5. Order of Testing Features

How Decision Trees Work

Decision trees work by making a series of decisions based on the values of input features or attributes to arrive at a final decision or prediction. These decisions are represented in a hierarchical tree structure, where each internal node represents a decision point, and each leaf node represents a final outcome or prediction.

- Selecting the Best Attribute to Split:
  - The decision tree construction process starts at the root node, which contains the entire dataset.
  - At each internal node, the algorithm selects the best attribute (feature) and a corresponding threshold value to split the data into subsets. The goal is to minimize impurity (in classification) or reduce variance (in regression).
- Splitting the Data:
  - The dataset is divided into subsets based on the values of the selected attribute. Each subset represents a branch or path in the decision tree.
  - In a binary decision, there are typically two branches for each internal node: one for instances where the condition is met and another for instances where it is not met.

- Assigning Class Labels (Classification) or Predictions (Regression):
  - Once the tree construction is complete, each leaf node is associated with a class label (in classification) or a predicted value (in regression).
  - When a new data instance is to be classified or predicted, it is passed through the decision tree, following the decision rules at each node.
  - The instance travels down the tree, making binary decisions at each internal node based on the attribute values, until it reaches a leaf node.
  - The class label or prediction at the leaf node is the final outcome.

1. Introduction
2. Key Concepts
3. How Decision Trees Work
4. Decision Tree Construction
5. Order of Testing Features

Example: Here is an example of using the emoji decision tree. Assume:

- I am 30 years old.
- This is work related.
- I am an accountant.
- I am not trying to get fired.

Following the tree, we answer no (not under 20 years old), no (not over 65 years old), yes (work related), no (not working in tech), and no (not trying to get fired). The leaf we reach is a thumb down, meaning we should not use emoji.

Problem

If we convert a decision tree to a program, what does it look like?

Solution

A decision tree corresponds to a program with a big nested if-then-else structure.

Example

Table 1: Decision Tree Example

Day	Outlook	Temperature	Humidity	Windy	Play Tennis
Day 1	Sunny	Hot	High	No	No
Day 2	Sunny	Hot	High	Yes	No
Day 3	Overcast	Hot	High	No	Yes
Day 4	Rainy	Mild	High	No	Yes
Day 5	Rainy	Cool	Normal	No	Yes
Day 6	Rainy	Cool	Normal	Yes	No
Day 7	Overcast	Cool	Normal	Yes	Yes
Day 8	Sunny	Mild	High	No	No
Day 9	Sunny	Cool	Normal	No	Yes
Day 10	Rainy	Mild	Normal	No	Yes
Day 11	Sunny	Mild	Normal	Yes	Yes
Day 12	Overcast	Mild	High	Yes	Yes
Day 13	Overcast	Hot	Normal	No	Yes
Day 14	Rainy	Mild	High	Yes	No

Problem: Construct a (full) decision tree for the Jeeves data set using the following order of testing features.

- First, test Outlook.
- For Outlook = Sunny, test Temp.
- For Outlook = Rain, test Wind.
- For other branches, test Humidity before testing Wind.

The construction of the tree is shown on page 9-11 from *Lecture_07_on_Decision_Trees.pdf*.
https://core.xjtlu.edu.cn/mod/folder/view.php?id=72712

When to stop splitting

- All samples for the given node belong to the same class
- No features left
- No samples left

1 Introduction

2 Key Concepts

3 How Decision Trees Work

4 Decision Tree Construction

5 Order of Testing Features

- Learning the simplest decision tree is an NP-complete problem
- Resort to a Greedy Approach
  - Start from empty decision tree
  - Split on the next best attribute (Feature)
  - Recurse

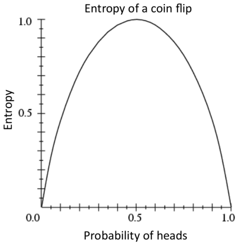
Identifying the most important feature

We decided to choose a feature that helps us make a decision as soon as possible, that is, a feature that **reduces our uncertainty** at much as possible.

Entropy

The concept of entropy is often used to quantify the amount of uncertainty or randomness in a dataset.
(Also called **entropy impurity**)

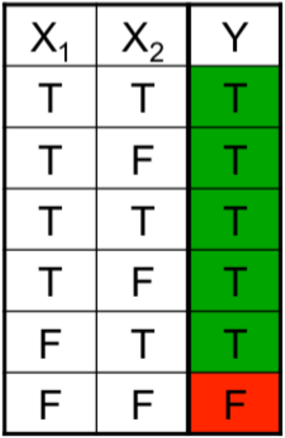
$$H(y) = -\sum_{i=1}^{k} P(y = y_i) \log_2 P(y = y_i)$$



More uncertainty, more entropy!
other metrics: Gini impurity

Information Theory interpretation: $H(Y)$is the expected number of bits needed to encode a randomly drawn value of Y (under most efficient code)

Entropy Example



X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F

$$P(Y = T) = \frac{5}{6}$$

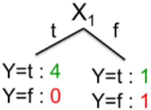
$$P(Y = F) = \frac{1}{6}$$

$$H(Y) = -\frac{5}{6}\log_2\frac{5}{6} - \frac{1}{6}\log_2\frac{1}{6} = 0.65$$

Conditinal Entropy

Conditional Entropy $H(y|x)$ of a random variable y conditioned on a random variable x

$$H(y|x) = -\sum_{j=1}^{m} P(x = x_j) \sum_{i=1}^{k} P(y = y_i|x = x_j) \log_2 P(y = y_i|x = x_j)$$



X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F

$$P(X = T) = \frac{4}{6}$$

$$P(X = F) = \frac{2}{6}$$

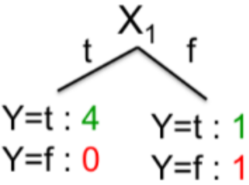
$$P(Y = T|X = T) = \frac{4}{4} = 1$$

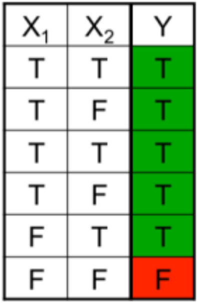
$$P(Y = F|X = T) = 0$$

$$P(Y = T|X = F) = \frac{1}{2}$$

$$P(Y = F|X = F) = \frac{1}{2}$$

$$H(Y|X_1) = -\frac{4}{6}(1 \log_2 1 + 0) - \frac{2}{6}(\frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{2} \log_2 \frac{1}{2}) = \frac{1}{3}$$





X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F

Information Gain

Decrease in entropy (uncertainty) after spliting

$$IG(x) = H(y) - H(y|x)$$

In the example, the information gain is

$$IG(x_1) = H(y) - H(y|x_1) = 0.65 - 0.33 = 0.32$$

Learning decision trees

1. Start from empty decision tree
2. Split on next best altribute (feature)
3. Use, for example, information gain to select altribute:

$$\arg\max_i IG(x_i) = \arg\max_i H(y) - H(y|x_i)$$

4. Recurse

Overfitting

- Standard decision trees have no learning bias
- Training set error is always zero!(If there is no label noise)
    - Lots of variance
    - Must introduce some bias towards simpler trees
- Many strategies for picking simpler trees
    - Use tricks
        - Fixed depth/Early stopping
        - Pruning
    - use ensembles of different trees (random forests)

Summary

- Decision trees are one of the most popular ML tools
- Easy to understand, implement, and use
- Computationally cheap
- Information gain to select alributes (ID3, C4.5,. . .)
- Presented for classification, can be used for regression
- Decision trees will overfit.

Questions

1. What is the entropy of the examples before we select a feature for the root node of the tree?
2. What is the information gain if we select Outlook as the root node of the tree?

Thank You !

Q & A