Decision Trees – Part 1

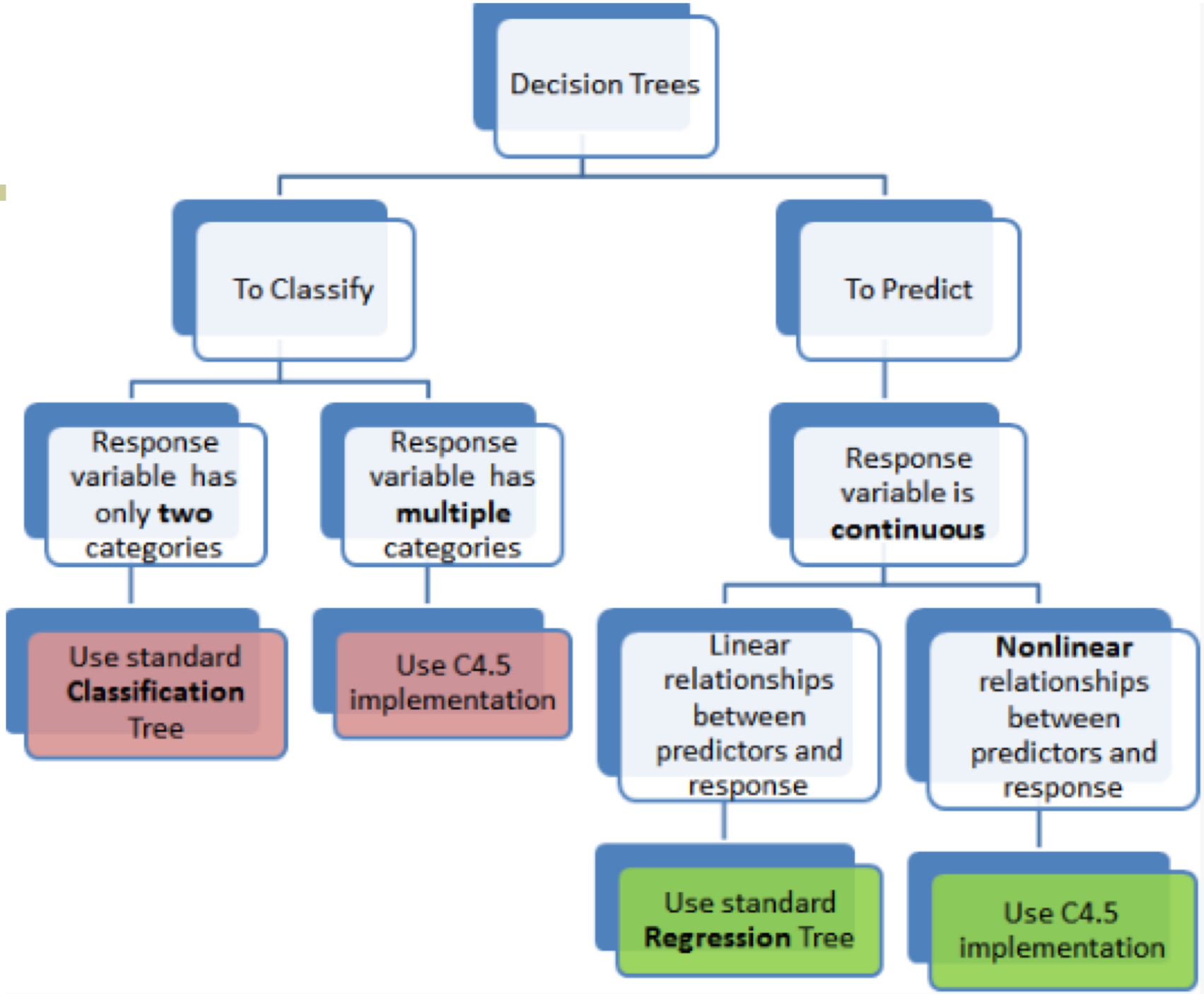
Rao Vemuri
University of California, Davis

Overview

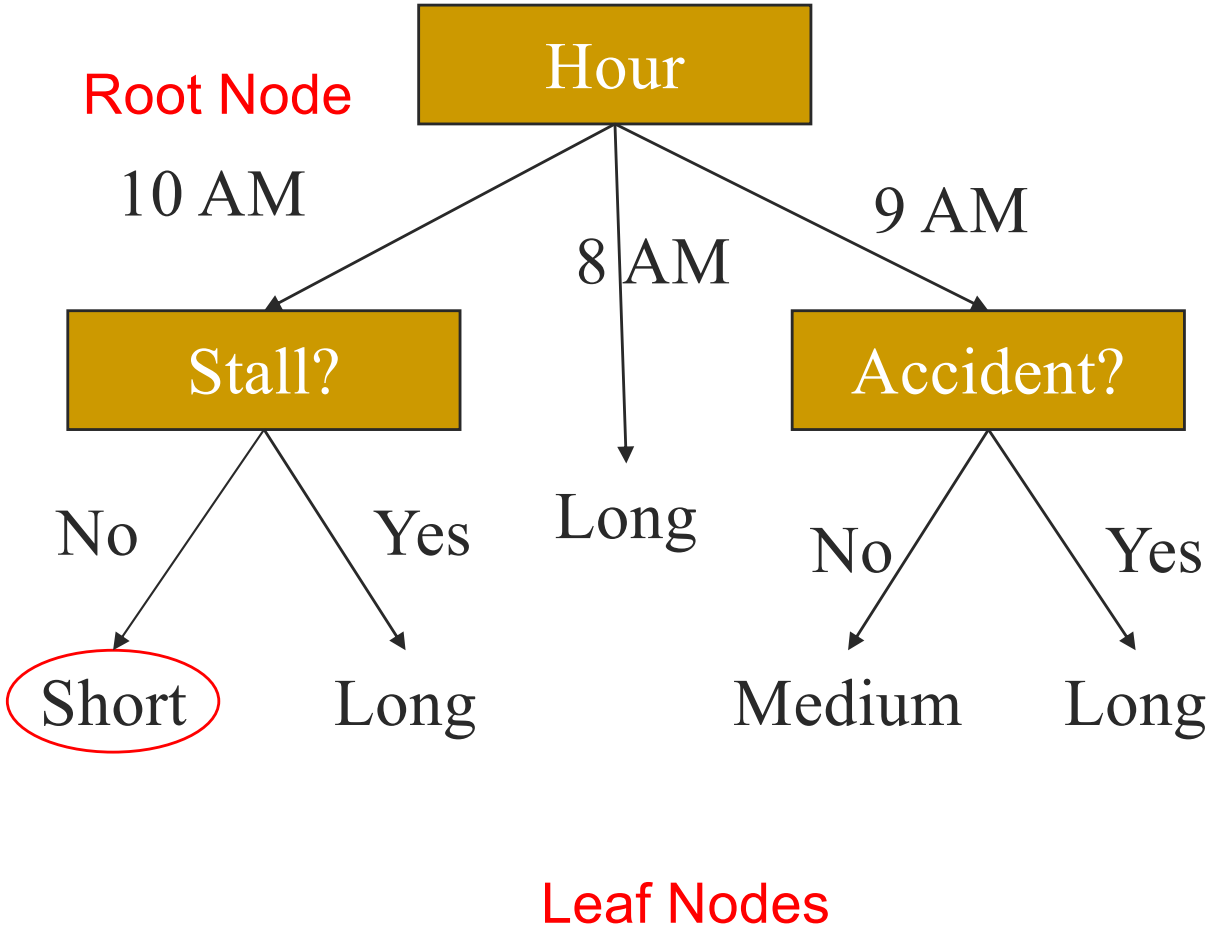
- What is a Decision Tree
- A Sample Decision Tree
  - Predicting Commute Time
- How to Construct a Decision Tree
- ID3 Algorithm (Entropy measure)
- Example Calculation (Play Tennis)
- Home Work & Quiz

What is a Decision Tree?

- Decision Trees (and their extension Random Forests) are robust and easy-to-interpret machine learning algorithms for Classification and Regression tasks.
- Decision Tree Learning is a simple and fast way of learning a function that maps data **x** to outputs **y:**
  - **x** can be a mixture of categorical and numeric variables
  - **y** can be categorical for classification, or numeric for regression.


Decision Trees
To Classify
To Predict
Response variable has only **two** categories
Response variable has **multiple** categories
Response variable is **continuous**
Use standard **Classification** Tree
Use C4.5 implementation
Linear relationships between predictors and response
Nonlinear relationships between predictors and response
Use standard **Regression** Tree
Use C4.5 implementation

Example: Predicting Commute Time



If we leave home at 10 AM & there are no cars stalled on the road, then commute time => Short

Inductive Learning

- In this decision tree, we made a series of Boolean decisions and followed the corresponding branch
  - Did we leave at 10 AM? Yes/No
  - Did a car stall on the road? Yes/No
  - Is there an accident on the road? Yes/No

- By answering each of these YES/NO questions, we came to a conclusion on how long our commute might take

Constructing a Decision Tree

- We first make a list of attributes that we can measure
  - These attributes (for now) must be discrete, categorical or Boolean
- We then choose a *target attribute* that we want to predict
- Then create an *experience table* that lists what we have seen in the past

Start with Experience Table

- Creating Experience Table is the first step for any machine learning method. It is your data.

Experience Table – Commute Time Dataset

Each row is an instance

Example	Attributes				Target
	Hour	Weather	Accident	Stall	Commute
D1	8 AM	Sunny	No	No	Long
D2	8 AM	Cloudy	No	Yes	Long
D3	10 AM	Sunny	No	No	Short
D4	9 AM	Rainy	Yes	No	Long
D5	9 AM	Sunny	Yes	Yes	Long
D6	10 AM	Sunny	No	No	Short
D7	10 AM	Cloudy	No	No	Short
D8	9 AM	Rainy	No	No	Medium
D9	9 AM	Sunny	Yes	No	Long
D10	10 AM	Cloudy	Yes	Yes	Long
D11	10 AM	Rainy	No	No	Short
D12	8 AM	Cloudy	Yes	No	Long
D13	9 AM	Sunny	No	No	Medium

Choosing Attributes - 1

- The “Commute Time” experience table showed 4 attributes: *hour, weather*, *accident* and *stall*
- But the decision tree only showed 3 attributes: *hour, accident* and *stall*
- No *weather*. Why is that?

Choosing Attributes - 2

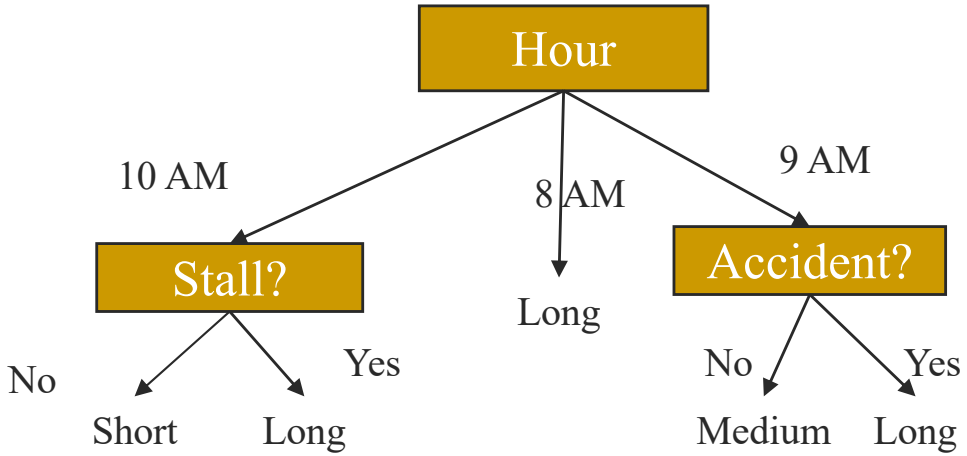
- Methods for selecting attributes (which will be described later) show that weather is not a *discriminating attribute*
  - That is, it doesn't matter what the weather is (in this example)

Decision Tree Algorithms

- The basic idea behind any decision tree algorithm is as follows:
  - Choose the *best* attribute to split the data and make that attribute a decision node
  - Repeat this process for each child node
  - Stop when:
    - All the instances have the same target attribute value
    - There are no more attributes
    - There are no more instances

Identifying the Best Attributes

- Refer back to our original decision tree



- How did we know to split the *Hour* first, then *stall* & *accident* and not consider *weather at all*?

Splitting Metrics

- Popular splitting metrics include the
  - maximizing the Information Gain (used by ID3, C4.5).
  - minimizing the Gini Impurity (used by CART)
  - We will consider ID3 in this lecture
  - (ID3: Iterative Dichotomizer 3)

ID3 Heuristic: Entropy

- To determine the best attribute to split, we look at the ID3 heuristic
- ID3 splits attributes based on their *entropy*.
- Entropy is a measure of “impurity” (or, non-homogeneity). A term borrowed from physics

Entropy (or Entropy Impurity)

- Entropy is minimized when all values of the target attribute are of the same class (homogeneous)
  - If we know that commute time will always be *short*, then entropy = 0
- Entropy is maximized when there is an equal chance for the target attribute to assume any value (i.e. the result is random)
  - If commute time = *short* in 3 instances, *medium* in 3 instances and *long* in 3 instances, then entropy = 1 (a maximum) (not homogeneous)

Entropy Formula in Boolean Classification

- Given a collection S, containing some positive and some negative examples of a target concept
  - Play Tennis? YES/NO
- In this case entropy of S is:

$$E(S) = -(p^+)\log_2(p^+) - (p^-)\log_2(p^-)$$

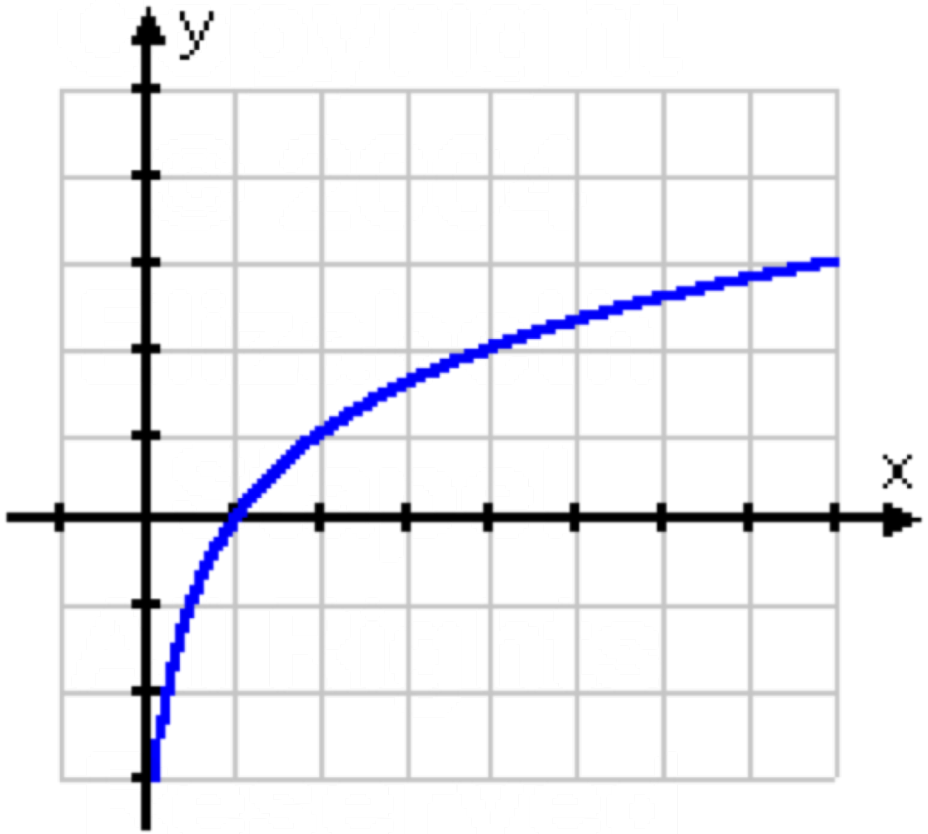
  - p+ = proportion in “positive class”
  - p^- = proportion in “negative class”

A Useful Formua for $\log_2$

- If you use a hand calculator, this formula will be useful

$$\log_2(x) = \frac{\ln(x)}{\ln(2)} = \frac{\log_{10}(x)}{\log_{10}(2)}$$

Graph of log



Review of log to base 2

- $\log_2(0)$ = (the formula is no good for a probability of 0)
- $\log_2(1) = 0$
- $\log_2(2) = 1$
- $\log_2(4) = 2$
- $\log_2(1/2) = -1$
- $\log_2(1/4) = -2$
- $(1/2)\log_2(1/2) = (1/2)(-1) = -1/2$

Play Tennis Dataset

Day	Outlook	Temp	Humidity	Wind	Tennis?
D1	Sunny	Hot	High	Weak	NO
D2	Sunny	Hot	High	Strong	NO
D3	Cloudy	Hot	High	Weak	YES
D4	Rain	Mild	High	Weak	YES
D5	Rain	Cool	Normal	Weak	YES
D6	Rain	Cool	Normal	Strong	NO
D7	Cloudy	Cool	Normal	Strong	YES
D8	Sunny	Mild	High	Weak	NO
D9	Sunny	Cool	Normal	Weak	YES
D10	Rain	Mild	Normal	Weak	YES
D11	Sunny	Mild	Normal	Strong	YES
D12	Cloudy	Mild	High	Strong	YES
D13	Cloudy	Hot	Normal	Weak	YES
D14	Rain	Mild	High	Strong	NO

Entropy of Play Tennis Data

- The dataset S has 14 instances: 9 positive, 5 negative: Use the notation S[9+, 5-]
- Entropy S[9+,5-] = - (9/14) $\log_2$(9/14)
  – (5/14) $\log_2$(5/14)
  = 0.940 (verify!)

Entropy Impurity = 0.940 (high!)

Information Gain

- How do we use the entropy idea to build a decision tree?
- Ask the question: “How much *entropy reduction* is achieved by partitioning the data set on a given attribute A?”
- Call this entropy (impurity) reduction by the name *Information Gain* (S, A)

Formula for Gain (S,A)

- Gain (S, A) = Entropy of the original dataset S – Expected value of the entropy after the dataset S is partitioned using attribute A:

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

where S_v is the subset of S for which A has a value v and |.| = cardinality

Example Calculation

- Consider the attribute *wind*. It has two values: *strong (6 rows), weak (8 rows)*
- When wind = *weak*, 6 of the examples are + and 2 are –
- S_{weak}= [6+, 2-]
- Similarly, S_{strong} = [3+, 3-]
- (Refer back to the dataset S)

Gain Calculation

- Gain (S, wind) = E(S) – (8/14) E(S_{weak}) – (6/14) E(S_{strong})

  = 0.940 – (8/14) 0.811 – (6/14) 1.00

  = 0.048 (verify!)

(Look back at the formula for Gain)

- Similarly, calculate Gain (S, outlook), Gain (S, humidity), Gain (S, Temp) and pick the attribute that gives the best gain

Home Work

- For the Play Tennis data set, calculate Gain (S, Humidity) and Gain (S, Wind) Gain (S, Outlook), and Gain (Temp)
- Using these gains, tell what attribute should be used to split the table
- Using the above attribute as the root node, draw a partial decision tree
- At this stage, are there any *leaf* nodes?
- Solve this problem, first (a) using paper, pencil, and hand calculator.

Home Work (contd)

- Repeat the steps until the entire decision tree is developed, using paper, pencil and hand calculator
- (b) Now solve the problem once again by writing code in Python or Scikit-Learn and compare the answers. Did you figure out how to display the decision tree?

When to Stop Splitting - 1

- If we continue to grow the tree fully until each leaf node corresponds to the lowest impurity, then the data have typically been overfit; in the limit, each leaf node has only one pattern!

When to Stop Splitting - 2

- If splitting is stopped too early, error on training data is not sufficiently low and performance will suffer
- Validation and cross-validation – Continue splitting until error on validation set is minimum – Cross-validation relies on several independently chosen subsets

When to Stop Splitting - 3

- Stop splitting when the best candidate split at a node reduces the impurity by less than the preset amount (threshold)
- How to set the threshold? Stop when a node has small no. of points or some fixed percentage of total training set (say 5%)

Summary

- Decision trees can be used to help predict the future
- The trees are easy to understand
- Decision trees work more efficiently with discrete attributes
- The trees may suffer from error propagation

Quiz 2 (online)

- Suppose that X1, ..., Xm are categorical input attributes and Y is categorical output attribute. Suppose we plan to learn a decision tree without pruning, using the standard algorithm.
- (True or False & Why): The maximum depth of the decision tree must be less than m+1.

Worked out Example: (Practice Session)

- **Build a decision tree to classify the following**

(x_1,x_2,x_3)	Class Label
(0, 0, 0)	0
(0, 0, 1)	0
(0, 1, 0)	0
(0, 1, 1)	0
(1, 1, 0)	0
(1, 0, 1)	1
(1, 1, 0)	0
(1, 1, 1)	1

Before you see the solution....

- Try to solve this problem in the class after lunch break.
- Then, verify your solution

Solution: Step 1

- 6 rows are Class 0; 2 rows are Class 1
- The initial entropy of all 8 points is
- $-(6/8)\log_2(6/8) - (2/8)\log_2(2/8) = 0.81$
- Suppose we divide the points in half by drawing a plane parallel to the x_2-x_3 plane. Then the left-branch has 4 points all belonging to the same class. So the entropy of the left branch is
- $-(4/4)\log_2(4/4) - (0/4)\log_2(0/4) = 0$

Solution: Step 2

- The right branch has two members of each class. The uncertainty of the right branch is
- $-(2/4)\log_2(2/4) - (2/4)\log_2(2/4) = 1$
- Average entropy after the first test (on x_1) is
- $(1/2)\,0 + (1/2)\,(1) = ½$
- Entropy reduction achieved is
- $0.81 - 0.5 = 0.31$

Solution: Step 3

- Do a similar thing along x_2 and x_3
- You will find out that test along x_3 gives exactly the same entropy and a test along x_2 gives no improvement at all. So first choose either x_1 or x_2.
- (Do the rest of the calculations!)
- The decision tree really implements f = x_1x_3.

By Inspection

- We could have decided on x1 or x2 by inspection alone, without any entropy calculation. Can you guess why?

Example (2-D Dataset)

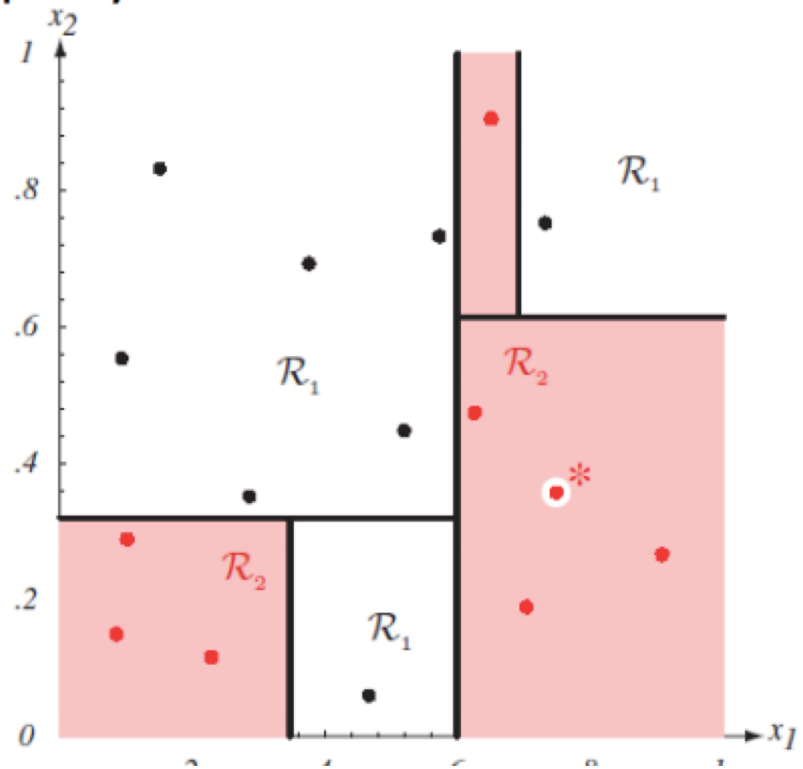
Consider the following tree, n = 16, in two dimensions for training a binary (CART) Tree using the entropy impurity

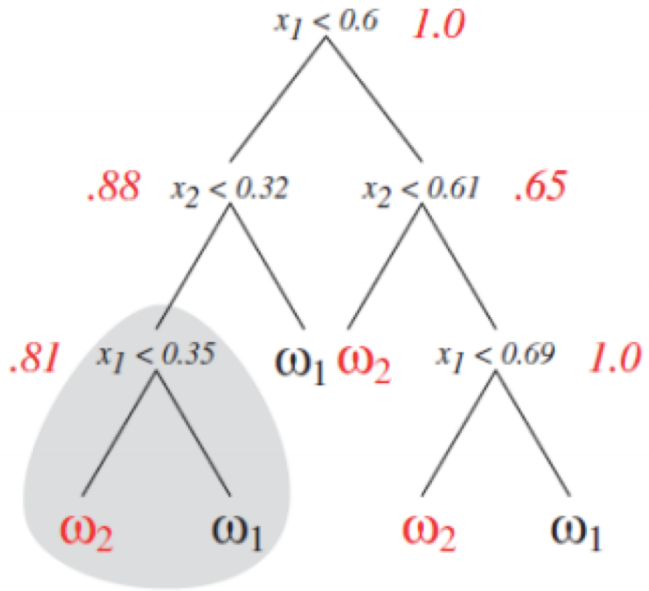
ω_1 (black)	
x_1	x_2
.15	.83
.09	.55
.29	.35
.38	.70
.52	.48
.57	.73
.73	.75
.47	.06

ω_2 (red)	
x_1	x_2
.10	.29
.08	.15
.23	.16
.70	.19
.62	.47
.91	.27
.65	.90
.75	.36* (.32†)

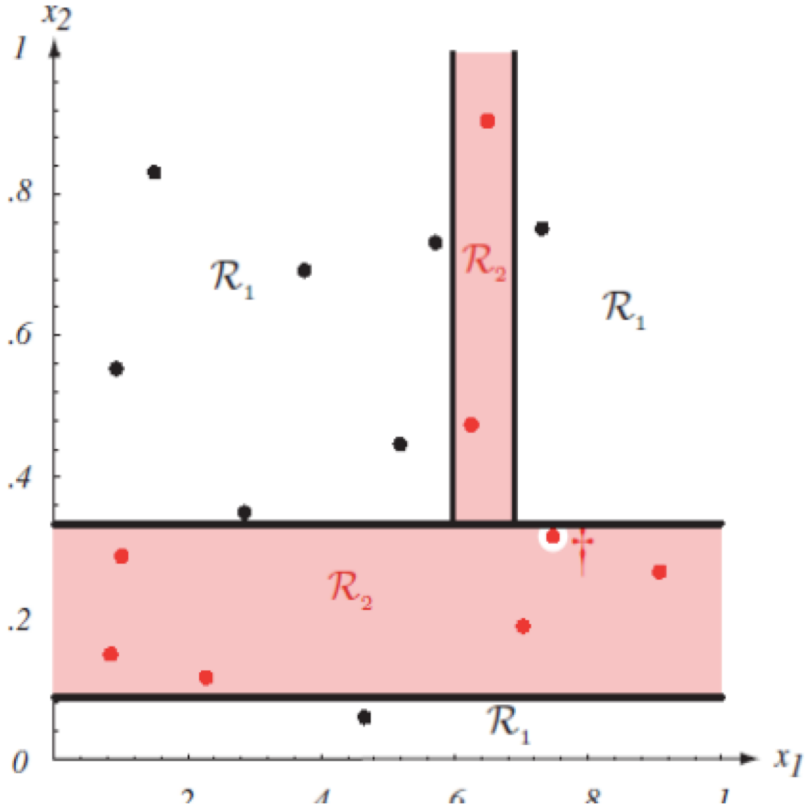
Example: Decision Tree

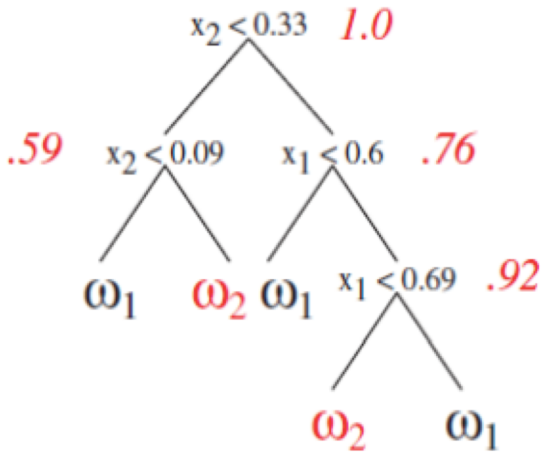
Entropy impurity at nonterminal nodes is shown in red and impurity at each leaf node is 0





Alternative Solution





Instability or sensitivity of tree to training points; alteration of a single point leads to a very different tree; due to discrete & greedy nature of CART

Quiz 2 (online)

- Suppose that X1, ..., Xm are categorical input attributes and Y is categorical output attribute. Suppose we plan to learn a decision tree without pruning, using the standard algorithm.
- (True or False & Why): The maximum depth of the decision tree must be less than m+1.

End of DT Part 1

- This is the end of the formal lecture.
- The rest of the material is for your reading