ELEC 515
Information Theory

Classification Using Decision Trees

How Do We Classify?

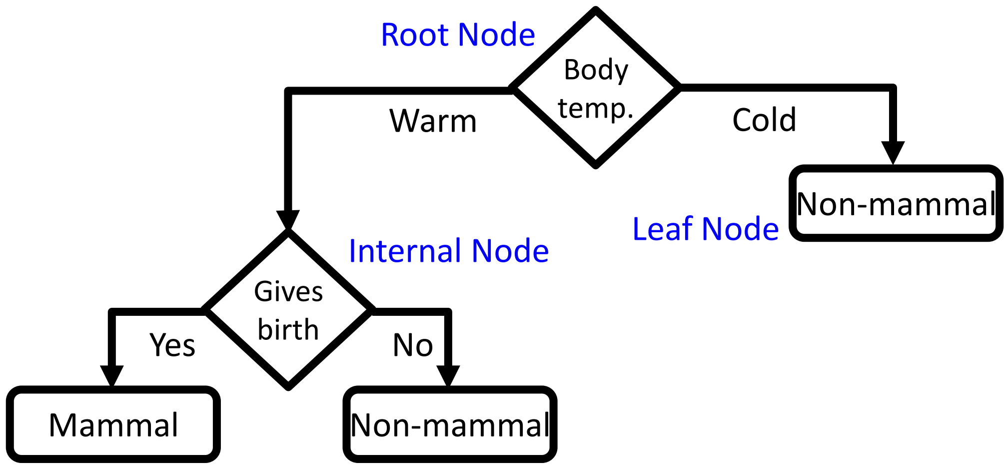
- Let's say we want to classify the climate in a region
- We may say something like
  - If the summer temperature is above 35 C for more than 20 days during May to July, the climate is tropical
  - If the total rainfall is less than 10 cm all year, the climate is desert
- Basic principles
  - Think of simple rules that place thresholds on some measurable features (attributes)
  - Combine rules to determine the classification

Decision Trees

- Decision trees are a popular and effective technique for solving classification problems
- In this method, the training data is broken down into smaller and smaller subsets in developing the tree
- At the end of the learning process, the tree is used for prediction

Decision Trees

- A Decision Tree (DT) defines a hierarchy of rules to make a prediction

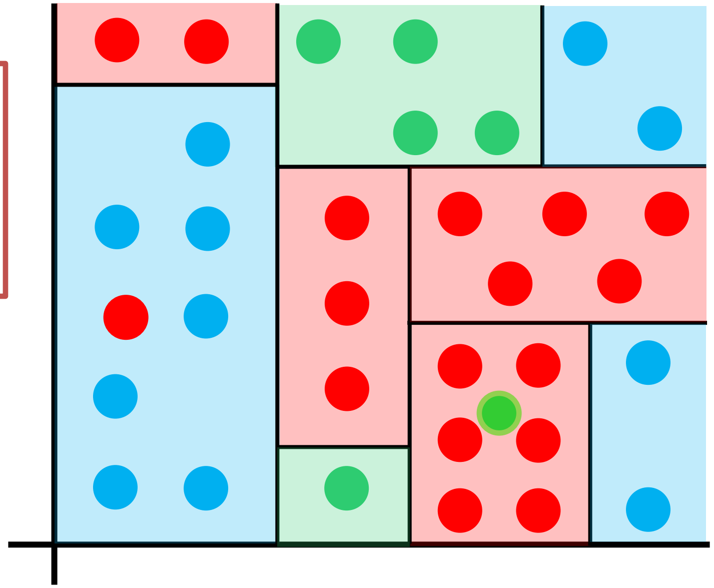


- Root and internal nodes test rules
- Leaf nodes make predictions

Decision Trees

- The basic idea is very simple
- Recursively partition the training data into homogeneous regions

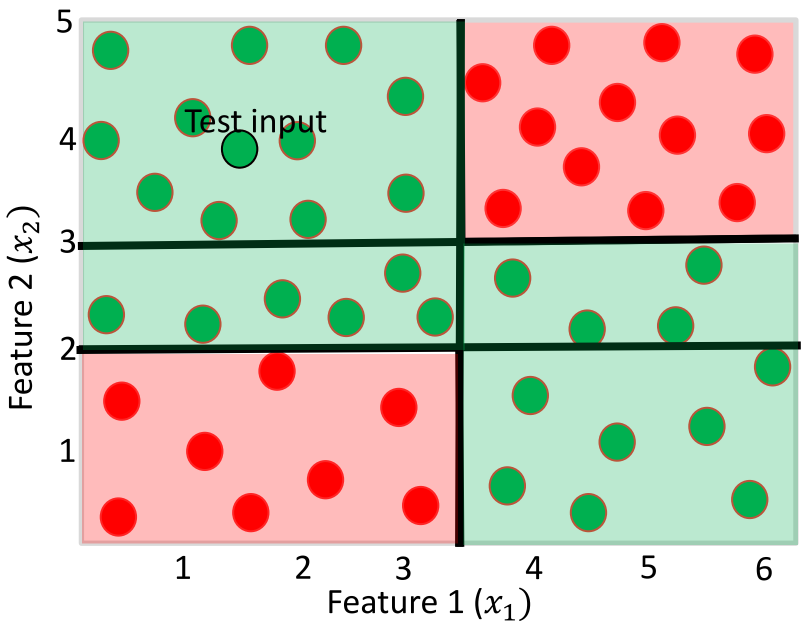
A homogeneous region will have all (or a majority of) training inputs with the same/similar outputs

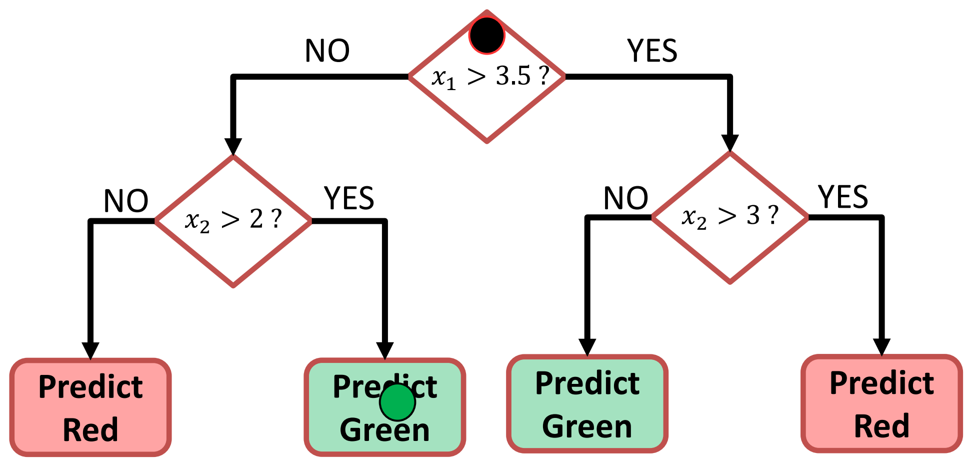


Even though the rules are simple, we can learn a fairly complex model
In this example, each rule is a simple horizontal/vertical classifier but the overall decision boundary is rather complex

- Within each group, predict the majority output/label

Decision Tree Classification





Testing with a DT is very efficient

The root node contains all training inputs
Each leaf node receives a subset of these inputs

Constructing Decision Trees

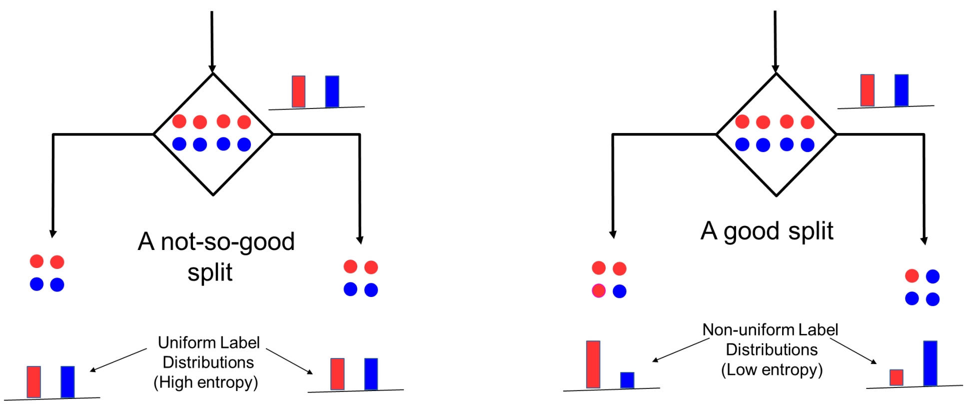
- Given some training data, what is the optimal DT?
- In general, constructing an optimal DT is an intractable problem (NP-hard)
- Often greedy heuristics are used to construct a good DT
- To do so, we use training data to determine which rules should be tested at each node
- The same rules will be applied to the test inputs to route them along the tree until they reach a leaf node where the prediction is made

Constructing Decision Trees

- How to decide which rules to test for and in what order?
- The rules should be organized such that the most informative rules are tested first
  - Informativeness of a rule is related to the purity of the split due to that rule
  - More informative rules yield more pure splits

How to Split at Nodes?

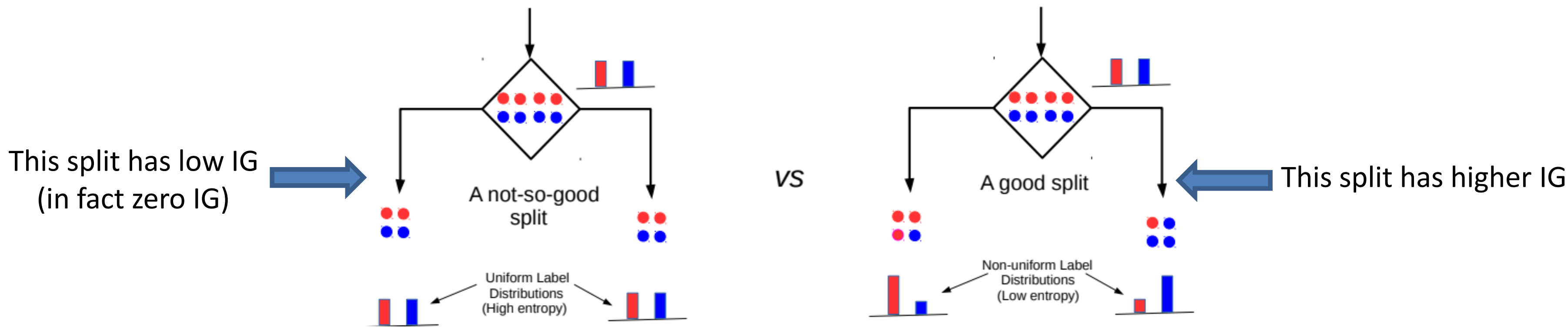
- Regardless of the rule, the split should result in as pure groups as possible
  - The majority of the training inputs should have the same label/output



- For classification problems (discrete outputs), entropy is a measure of purity
  - Low entropy ⇒ high purity (less uniform label distribution)
  - Splits that give the largest reduction (before split versus after split) in entropy are preferred (this reduction is called information gain)

Information Gain

- Consider a set of inputs S
- Suppose a rule splits S into two disjoint sets S_1 and S_2 based on a feature F
- The reduction in entropy after the split is the Information Gain $IG = H(S) - H(S|F) = I(S;F)$



DT Classification Example

- Deciding whether to play or not to play tennis on a Saturday (binary classification)
- There are 4 features
  - Outlook (O)
  - Temperature (T)
  - Humidity (H)
  - Wind (W)
- Each internal node will test the value of one of the features

Training Data

day	Outlook (O)	Temperature (T)	Humidity (H)	Wind (W)	Play (P)
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

Information Gain

$$IG = I(Y;X) \quad = \quad H(Y) - H(Y|X)$$

$$\begin{aligned} H(Y|X) &= -\sum_{i=1}^{N}\sum_{j=1}^{M} p(x_i, y_j) \log p(y_j|x_i) \\ &= -\sum_{i=1}^{N}\sum_{j=1}^{M} p(x_i)p(y_j|x_i) \log p(y_j|x_i) \\ &= -\sum_{i=1}^{N} p(x_i) \sum_{j=1}^{M} p(y_j|x_i) \log p(y_j|x_i) \end{aligned}$$

Training Data

day	Outlook (O)	Temperature (T)	Humidity (H)	Wind (W)	Play (P)
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

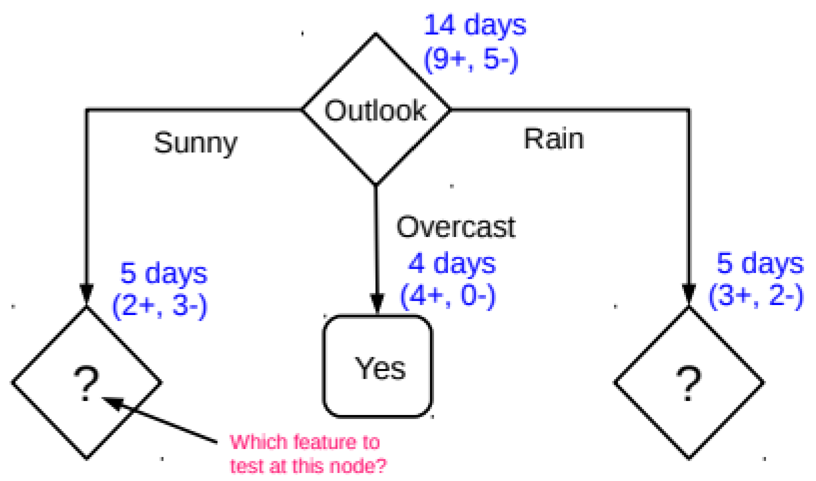
- At the root node, compute the IG for all 4 features

$$H(P) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) = 0.940 \text{ bit}$$

- For Wind the conditional entropy is

$$H(P|W) = p(weak)[-p(no|weak)\log_2 p(no|weak)$$
$$-p(yes|weak)\log_2 p(yes|weak)]$$
$$+p(strong)[-p(no|strong)\log_2 p(no|strong)$$
$$-p(yes|strong)\log_2 p(yes|strong)]$$
$$= 8/14[-2/8\log_2 2/8-6/8\log_2 6/8]+6/14[-3/6\log_2 3/6-3/6\log_2 3/6]$$
$$= .892 \text{ bit}$$

- $I(P;W) = H(P) - H(P|W)$
  $= .940 - .892 = .048$ bit
- $I(P;O) = H(P) - H(P|O)$
  $= .940 - .694 = .246$ bit
- $I(P;T) = H(P) - H(P|T)$
  $= .940 - .911 = .029$ bit
- $I(P;H) = H(P) - H(P|O)$
  $= .940 - .788 = .152$ bit
- Outlook provides the greatest IG


14 days
(9+, 5-)
Outlook
Sunny
Rain
Overcast
5 days
(2+, 3-)
4 days
(4+, 0-)
5 days
(3+, 2-)
?
Yes
?
Which feature to
test at this node?

Training Data

day	Outlook (O)	Temperature (T)	Humidity (H)	Wind (W)	Play (P)
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rain	mild	high	weak	yes
5	rain	cool	normal	weak	yes
6	rain	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rain	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rain	mild	high	strong	no

Growing the Tree

- At the sunny node, compute the IG for the 3 remaining features

$$H(P) = -(3/5)\log_2(3/5) - (2/5)\log_2(2/5) = 0.971 \text{ bit}$$

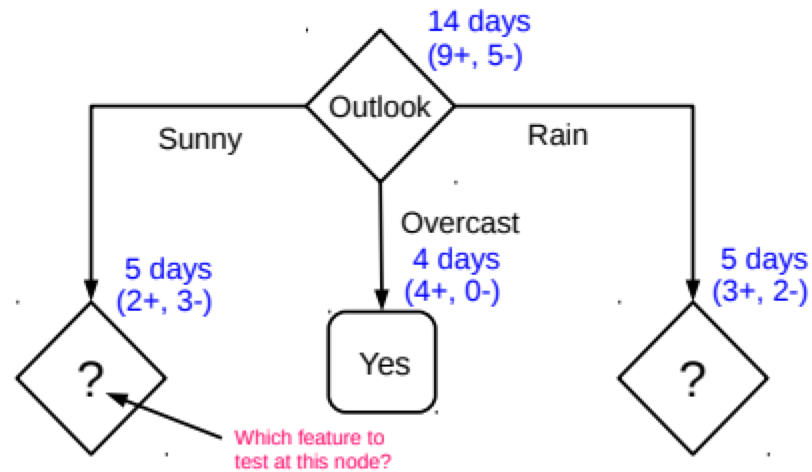
- For Temperature the conditional entropy is

$$
\begin{aligned}
H(P|T) = &\ p(hot)[-p(no|hot)\log_2 p(no|hot) \\
&\qquad -p(yes|hot)\log_2 p(yes|hot)] \\
&+p(mild)[-p(no|mild)\log_2 p(no|mild) \\
&\qquad -p(yes|mild)\log_2 p(yes|mild)] \\
&+p(cool)[-p(no|cool)\log_2 p(no|cool) \\
&\qquad -p(yes|cool)\log_2 p(yes|cool)]
\end{aligned}
$$

$$= 2/5[-1\log_2 1-0\log_2 0]+2/5[-1/2\log_2 1/2-1/2\log_2 1/2]+1/5[-0\log_2 0-1\log_2 1]$$

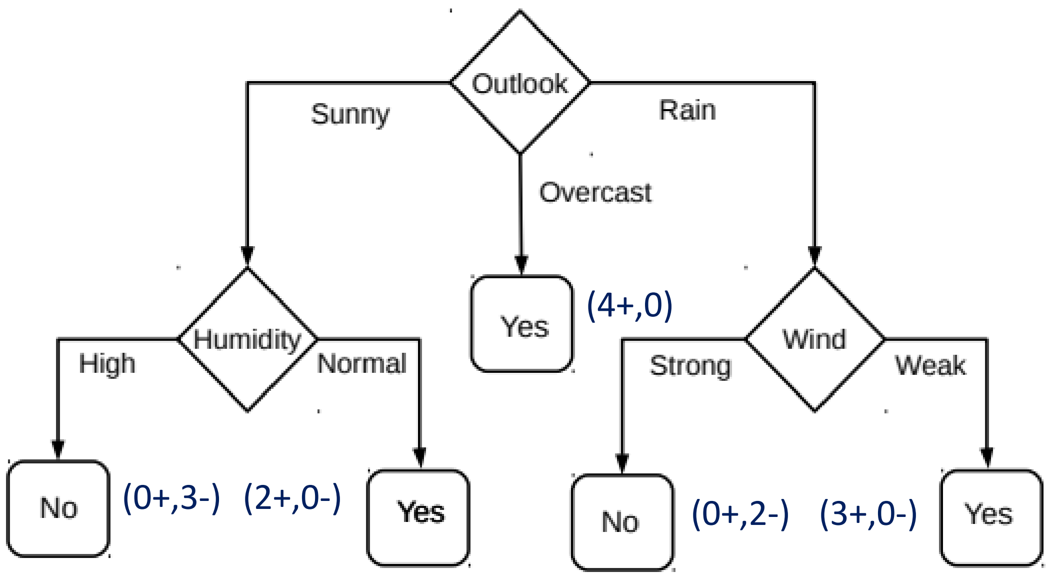
$$= .400 \text{ bit}$$

Growing the Tree



- Proceeding as before, for level 2
  - Left node:
    - I(P;T) = 0.571
    - I(P;H) = 0.971
    - I(P;W) = 0.020
    - Choose humidity as the feature to be tested
  - Middle node: no need to expand as it is pure (all training data have output yes)
  - Right node: Wind has the largest IG
- Note that if a feature has already been tested on a path, don't consider it again

Final Tree



Test data: Sunny, Cool, High, Weak

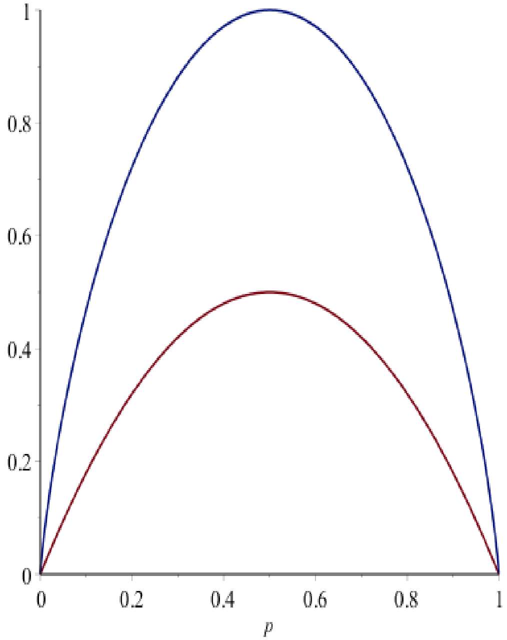
Rain, Mild, Normal, Strong

Gini Index

- The Gini index is defined as

$$Gini = \sum_{i=1}^{N} p_i(1-p_i) = \sum_{i=1}^{N} p_i - \sum_{i=1}^{N} p_i^2 = 1 - \sum_{i=1}^{N} p_i^2$$

- It is used as an alternative to entropy
- Gini is used in
  - Classification and Regression Tree (CART)
- Entropy is used in
  - Iterative Dichometer 3 (ID3)
  - C4.5 (descendent of ID3)



H(X) versus Gini for *N*=2