Attribute selection measure: Gain ratio

- Information gain is biased towards attributes with a large number of values
  - Consider the attribute ID (unique identifier)
- C4.5 (a successor of ID3) uses gain ratio to overcome the problem, which normalizes the gain by split information:

$$\text{GainRatio}(S, A) = \frac{\text{Gain}(S, A)}{\text{SplitInfo}(S, A)}$$

Measures the information w.r.t. classification

Measures the information generated by splitting S into |Values(A)| partitions

$$\text{SplitInfo}(S, A) = -\sum_{v \in Values(A)} P_v \bullet \log_2(P_v) = -\sum_{v \in Values(A)} \frac{|S_v|}{|S|} \bullet \log_2\left(\frac{|S_v|}{|S|}\right)$$

  - High split info: many partitions that have more or less the same size (uniform)
  - Low split info: few partitions hold most of the tuples (peaks)
  - If an attribute produces many splits → high SplitInfo()→low GainRatio().
- The attribute with the maximum gain ratio is selected as the splitting attribute

Example: Split information

- Example:
  - Humidity={High, Low}

$$SplitInformation(S, Humidity) = -\frac{7}{14}\times\log_2(\frac{7}{14}) - \frac{7}{14}\times\log_2(\frac{7}{14}) = 1$$

  - Wind={Weak, Strong}

$$SplitInformation(S, Wind) = -\frac{8}{14}\times\log_2(\frac{8}{14}) - \frac{6}{14}\times\log_2(\frac{6}{14}) = 0.9852$$

  - Outlook = {Sunny, Overcast, Rain}

$$SplitInformation(S, Outlook) = -\frac{5}{14}\times\log_2(\frac{5}{14}) - \frac{4}{14}\times\log_2(\frac{4}{14}) - \frac{5}{14}\times\log_2(\frac{5}{14}) = 1.5774$$

Training set

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	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	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Attribute selection measure: Gini Index 1/2

- Used in CART
- Let a dataset S containing examples from k classes. Let p_j be the probability of class j in S. The Gini Index of S is given by:

$$\text{Gini}(S) = 1 - \sum_{j=1}^{k} p_j^2$$

  - Gini index considers a binary split for each attribute
- If S is split based on attribute A into two subsets S_1 and S_2 :

$$Gini(S,A) = \frac{|S_1|}{|S|} Gini(S_1) + \frac{|S_2|}{|S|} Gini(S_2)$$

- Reduction in impurity:

$$\Delta Gini(S,A) = Gini(S) - Gini(S,A)$$

- The attribute A that provides the smallest G*ini*(*S,A*) (or the largest reduction in impurity) is chosen to split the node

Attribute selection measure: Gini Index 2/2

- How to find the binary splits?
  - For discrete-valued attributes, we consider all possible subsets that can be formed by values of A (next slides)
  - For numerical attributes, we find the split points (next slides)

Gini index example for discrete-valued attributes 1/2

- Let *D* has 14 instances
  - 9 of class *buys_computer = "yes"*
  - 5 in *buys_computer = "no"*
- The Gini Index of *D* is:

$$\mathrm{Gini}(D) = 1 - \sum_{j=1}^{k} p_j^2 \Rightarrow \mathrm{Gini}(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

- Let the attribute "*Income*" = {*low, medium, high*} .
- To generate the binary splits for *"Income"*, we check all possible subsets:
  - (*{low,medium} and {high}*)
  - (*{low,high} and {medium}*)
  - (*{medium,high} and {low}*)

Gini index example for discrete-valued attributes 2/2

- For each subset, we check the Gini Index:
- For example, (*{low,medium} and {high}*) split result in D_1 (#10 instances) and D_2(#4 instances)

$$Gini_{\{low,medium\} and \{high\}}(D) = \left(\frac{10}{14}\right) Gini(D_1) + \left(\frac{4}{14}\right) Gini(D_2)$$

$$= \frac{10}{14}(1 - (\frac{6}{10})^2 - (\frac{4}{10})^2) + \frac{4}{14}(1 - (\frac{1}{4})^2 - (\frac{3}{4})^2)$$
$$= 0.450$$

- For the remaining binary split partitions:

$$Gini_{\{low, high\} and \{medium\}}(D) = 0.315$$
$$Gini_{\{medium, high\} and \{low\}}(D) = 0.300$$

- So, the best binary split for income is on *({medium, high} and {low})*

Dealing with continuous attributes 1/2

- Let attribute A be a continuous-valued attribute
- Must determine the *best split point* t for A, (A ≤ t)
  - Sort the value A in increasing order
  - Identify adjacent examples that differ in their target classification
    - Typically, every such pair suggests a potential split threshold t= $(a_i+a_{i+1})/2$
  - Select threshold t that yields the best value of the splitting criterion.

Temperature:	40	48	60	72	80	90
PlayTennis:	No	No	Yes	Yes	Yes	No

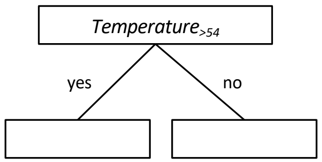
t=(48+60)/2=54 t =(80+90)/2=85

- 2 potential thresholds:Temperature$_{>54}$, Temperature $_{>85}$
- Compute the attribute selection measure (e.g. information gain) for both
- Choose the best (Temperature$_{>54}$ here)

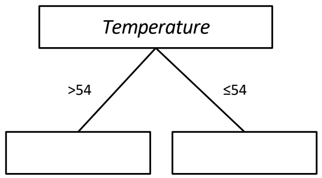
Dealing with continuous attributes 2/2

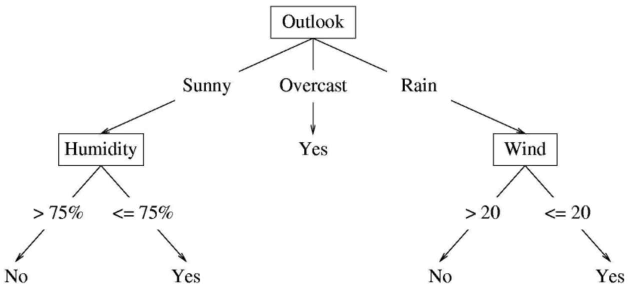
- Let t be the threshold chosen from the previous step
- Create a boolean attribute based on A and threshold t with two possible outcomes: yes, no
  - S_1 is the set of tuples in S satisfying (A >t), and S_2 is the set of tuples in S satisfying (A ≤ t)

How it looks



or



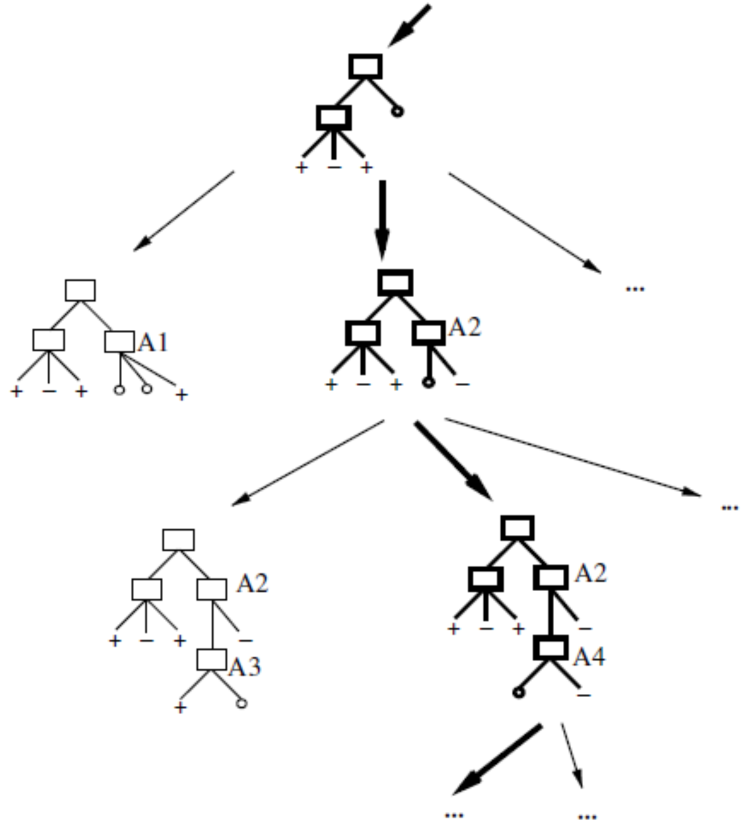


An example of a tree for the play tennis problem when attributes Humidity and Wind are continuous

Comparing Attribute Selection Measures

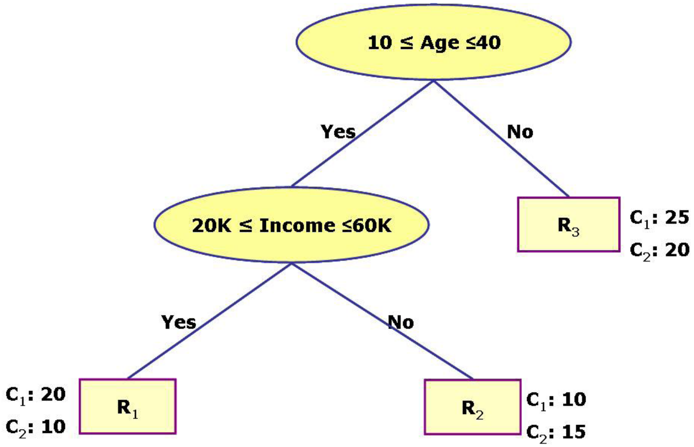
- The three measures, are commonly used and in general, return good results but
  - Information gain Gain(S,A):
    - biased towards multivalued attributes
  - Gain ratio GainRatio(S,A) :
    - tends to prefer unbalanced splits in which one partition is much smaller than the others
  - Gini index:
    - biased to multivalued attributes
    - has difficulty when # of classes is large
    - tends to favor tests that result in equal-sized partitions and purity in both partitions
- Several other measures exist

Hypothesis search space (by ID3)

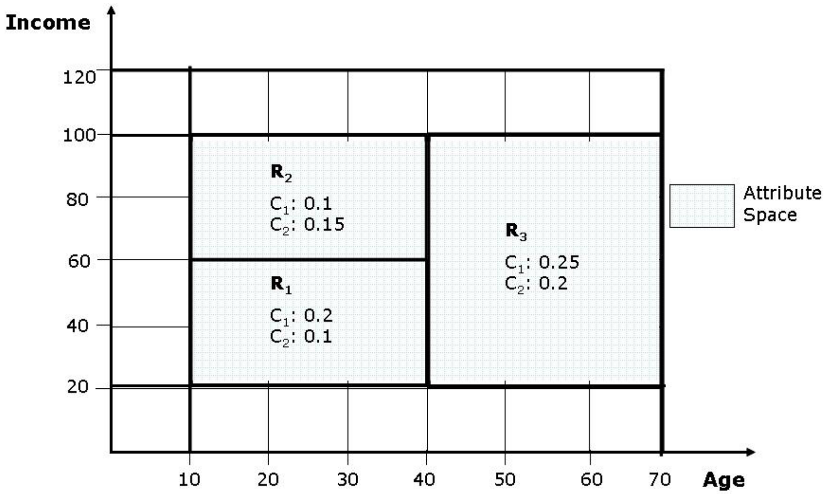


- Hypothesis space is complete
  - Solution is surely in there
- Greedy approach
- No back tracking
  - Local minima
- Outputs a single hypothesis

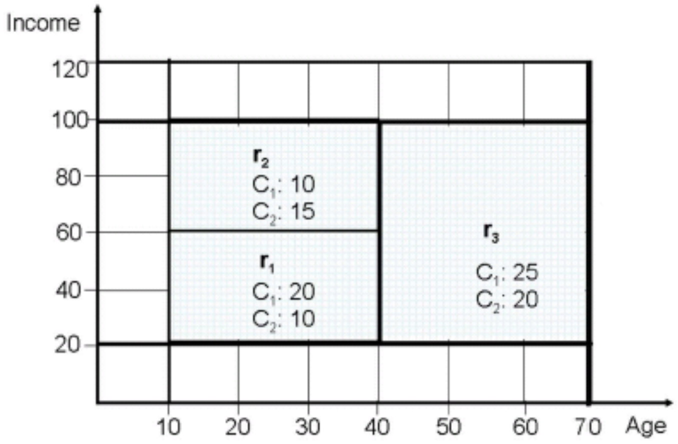
Partition-based methods



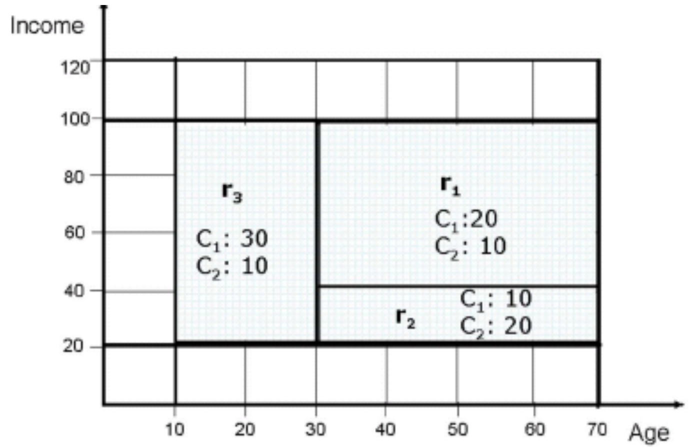
- DTs partition the space into rectangular regions
- Decision regions: axis parallel hyper-rectangles
- Decision boundary: the border line between two neighboring regions of different classes



Comparing DTs/ partitionings



DT1



DT2

When to consider decision trees

- Instances are represented by attribute-value pairs
  - Instances are represented by a fixed number of attributes, e.g. outlook, humidity, wind and their values, e.g. (wind=strong, outlook =rainy, humidity=normal)
  - The easiest situation for a DT is when attributes take a small number of disjoint possible values, e.g. wind={strong, weak}
  - There are extensions for numerical attributes also, e.g. temperature, income.
- The class attribute has discrete output values
  - Usually binary classification, e.g. {yes, no}, but also for more class values, e.g. {pos, neg, neutral}
- The training data might contain errors
  - DTs are robust to errors: both errors in the class values of the training examples and in the attribute values of these examples
- The training data might contain missing values
  - DTs can be used even when some training examples have some unknown attribute values

Outline

- Classification basics
- Decision tree classifiers
- Overfitting
- Lazy vs Eager Learners
- k-Nearest Neighbors (or learning from your neighbors)
- Evaluation of classifiers