Decision Trees

3 Jun 2019

Vineeth N Balasubramanian

आई आई टी हैदराबाद
IIT Hyderabad

Classification Methods

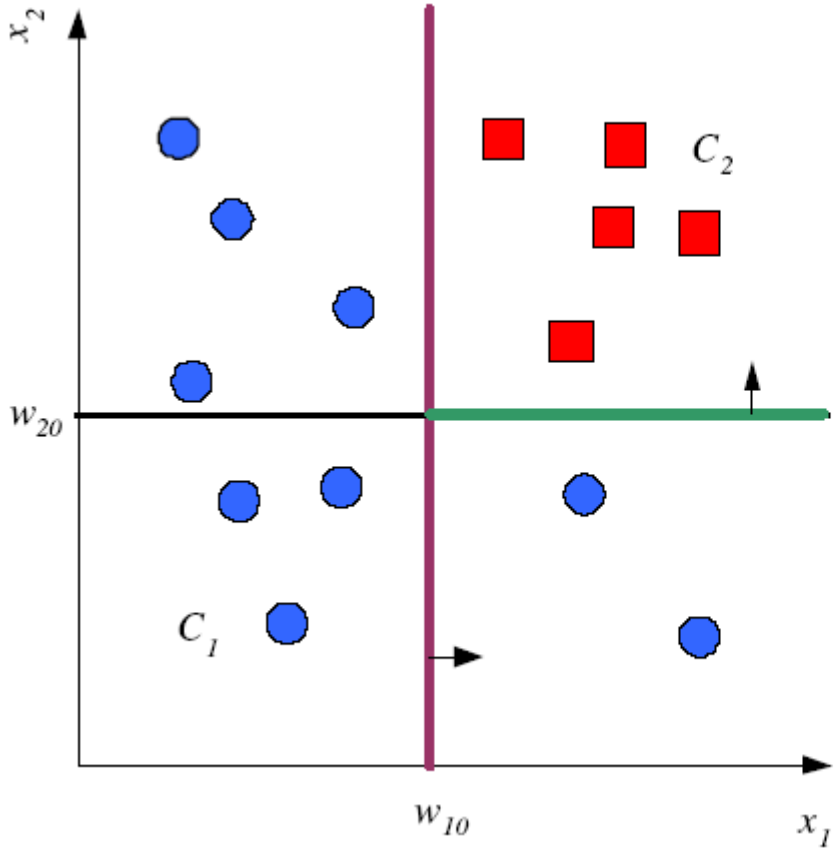
- k-Nearest Neighbors
- Decision Trees
- Naïve Bayes
- Support Vector Machines
- Logistic Regression
- Neural Networks
- Ensemble Methods (Boosting, Random Forests)

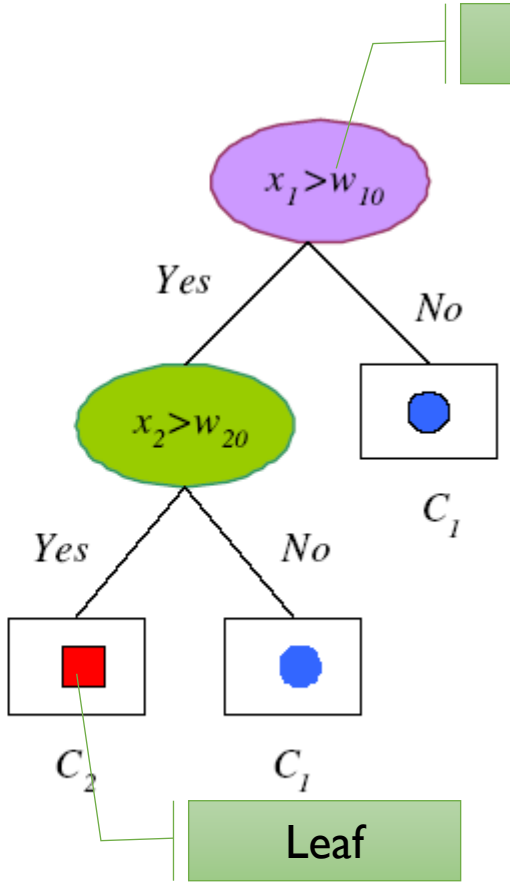
Example

PlayTennis: training examples

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

Decision Trees





- An efficient nonparametric method
- A hierarchical model
- Divide-and–conquer strategy

Source: Ethem Alpaydin, Introduction to Machine Learning, 3rd Edition (Slides)

Divide and Conquer

- Internal decision nodes
  - Univariate: Uses a single attribute, x_i
    - Numeric x_i :
      - Binary split : $x_i > w_m$
    - Discrete x_i :
      - n-way split for n possible values
  - Multivariate: Uses more than one attributes, $\boldsymbol{x}$
- Leaves
  - Classification: Class labels, or proportions
  - Regression: Numeric; r average, or local fit
- Learning is greedy; find the best split recursively

Source: Ethem Alpaydin, Introduction to Machine Learning, 3rd Edition (Slides)

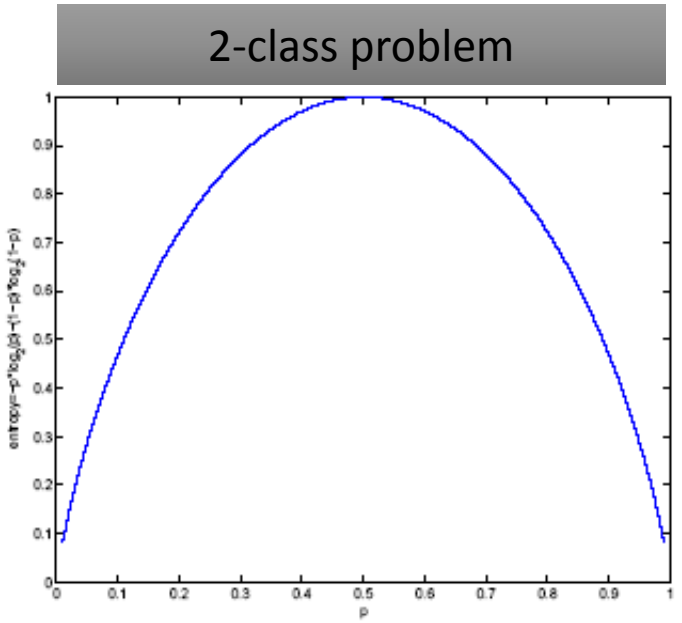
Classification Trees (C4.5, J48)

- For node m, N_m instances reach m, N^i_m belong to C_i

$$\hat{P}(C_i \mid \mathbf{x}, m) \equiv p^i_m = \frac{N^i_m}{N_m}$$

- Node m is **pure** if p^i_m is 0 or 1
- Measure of **impurity** is **entropy**

$$I_m = -\sum_{i=1}^{K} p^i_m \log_2 p^i_m$$



Entropy in information theory specifies the average (expected) amount of information derived from observing an event

Source: Ethem Alpaydin, Introduction to Machine Learning, 3rd Edition (Slides)

Classification Trees

- If node m is pure, generate a leaf and stop, otherwise split and continue recursively
- **Impurity after split:** N_{mj} of N_m take branch j. N^i_{mj} belong to C_i

$$\hat{P}(C_i \mid \mathbf{x}, m, j) \equiv p^i_{mj} = \frac{N^i_{mj}}{N_{mj}}$$

$$I'_m = -\sum_{j=1}^{n} \frac{N_{mj}}{N_m} \sum_{i=1}^{K} p^i_{mj} \log_2 p^i_{mj}$$

- **Information Gain:** Expected reduction in impurity measure after split
- Choose the *best* attribute(s) (**with maximum information gain**) to split the remaining instances and make that attribute a decision node
  - You can use same logic to find best splitting value too

Source: Ethem Alpaydin, Introduction to Machine Learning, 3rd Edition (Slides)

Other Measures of Impurity

- The properties of functions measuring the **impurity** of a split:
  - $\phi(1/2,1/2) \geq \phi(p,1-p),\ \text{for any}\ p \in [0,1]$
  - $\phi(0,1) = \phi(1,0) = 0$
  - $\phi(p,1-p)$ is increasing in p on $[0,\frac{1}{2}]$

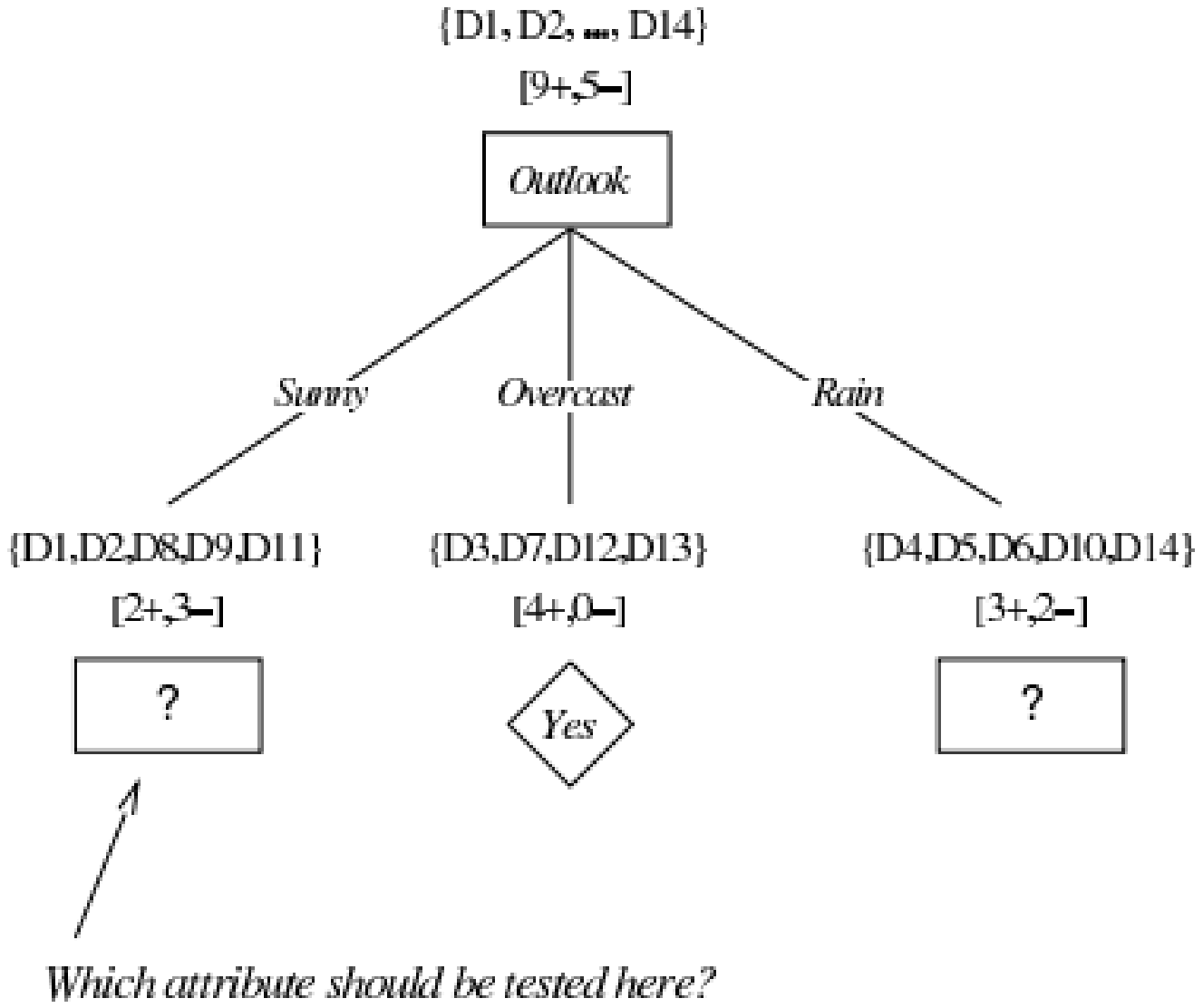
    and decreasing in p on $[\frac{1}{2},1]$
- Examples (other than entropy)
  - **Gini impurity/index:** $1 - \sum_{j=1}^{c} p_j^2$

Decision Trees: Example

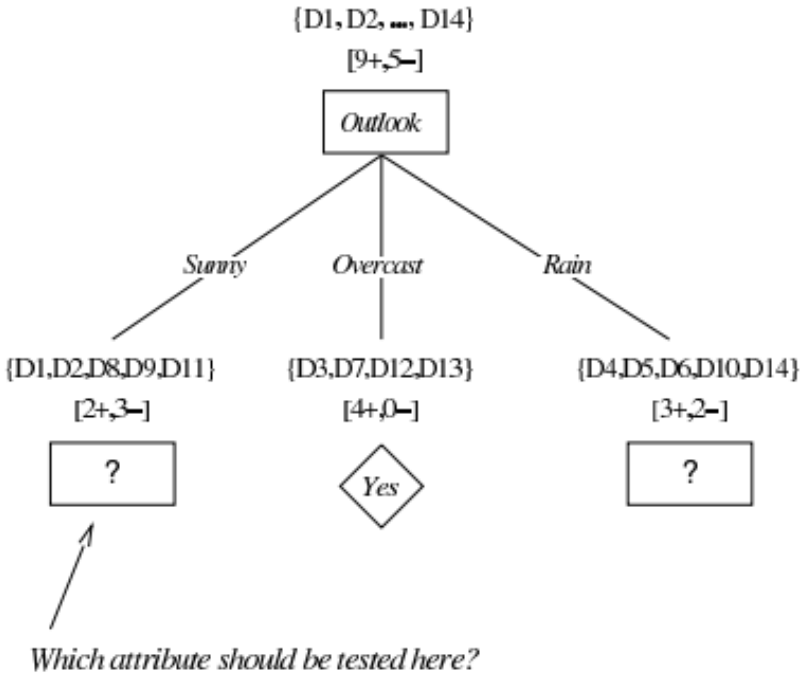
PlayTennis: training examples

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

Decision Trees: Example



Decision Trees: Example



$S_{sunny} = \{D1, D2, D8, D9, D11\}$

$Gain(S_{sunny}, Humidity) = .970 - (3/5)\,0.0 - (2/5)\,0.0 = .970$

$Gain(S_{sunny}, Temperature) = .970 - (2/5)\,0.0 - (2/5)\,1.0 - (1/5)\,0.0 = .570$

$Gain(S_{sunny}, Wind) = .970 - (2/5)\,1.0 - (3/5)\,.918 = .019$

Overfitting and Generalization

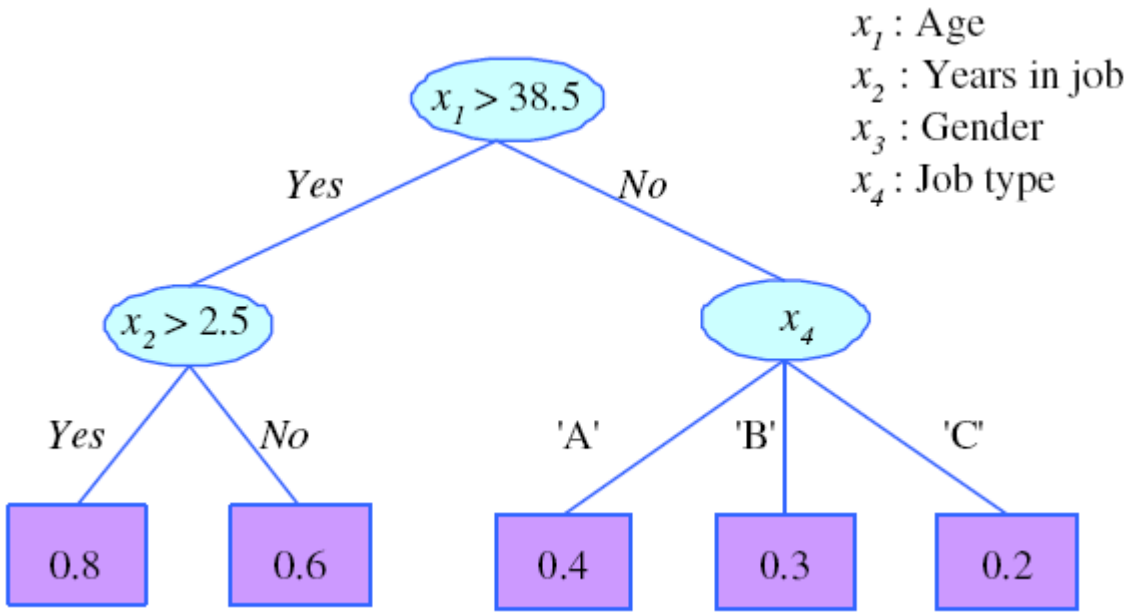
- Overfitting can occur with noisy training examples, also when small numbers of examples are associated with leaf nodes. How to handle?
- **Pruning:** Remove subtrees for better generalization (decrease variance)
  - Prepruning: Early stopping
  - Postpruning: Grow the whole tree then prune subtrees which overfit on the pruning set
  - Prepruning is faster, postpruning is more accurate

Overfitting and Generalization

- **Occam's Razor principle:** when multiple hypotheses can solve a problem, choose the simplest one
  - a short hypothesis that fits data unlikely to be coincidence
  - a long hypothesis that fits data might be coincidence

- How to select "best" tree:
  - Measure performance over training data
  - Measure performance over separate validation data set
  - **Minimum Description Length:** Minimize *size(tree)* + *size(misclassifications(tree))*

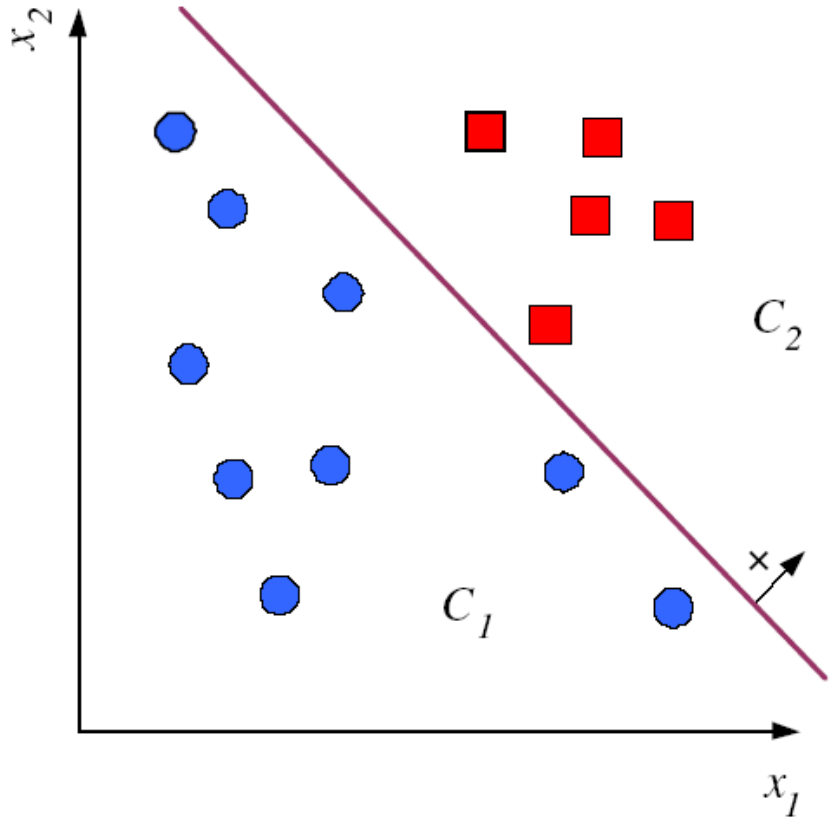
Rule Extraction from Trees

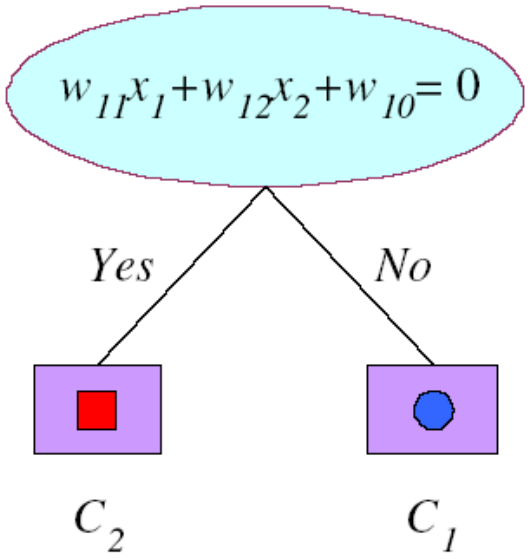


R1: IF (age>38.5) AND (years-in-job>2.5) THEN y =0.8
R2: IF (age>38.5) AND (years-in-job≤2.5) THEN y =0.6
R3: IF (age≤38.5) AND (job-type='A') THEN y =0.4
R4: IF (age≤38.5) AND (job-type='B') THEN y =0.3
R5: IF (age≤38.5) AND (job-type='C') THEN y =0.2

- Convert tree to equivalent set of rules
- Prune each rule independently of others, by removing any preconditions that result in improving its estimated accuracy
- Sort final rules into desired sequence for use

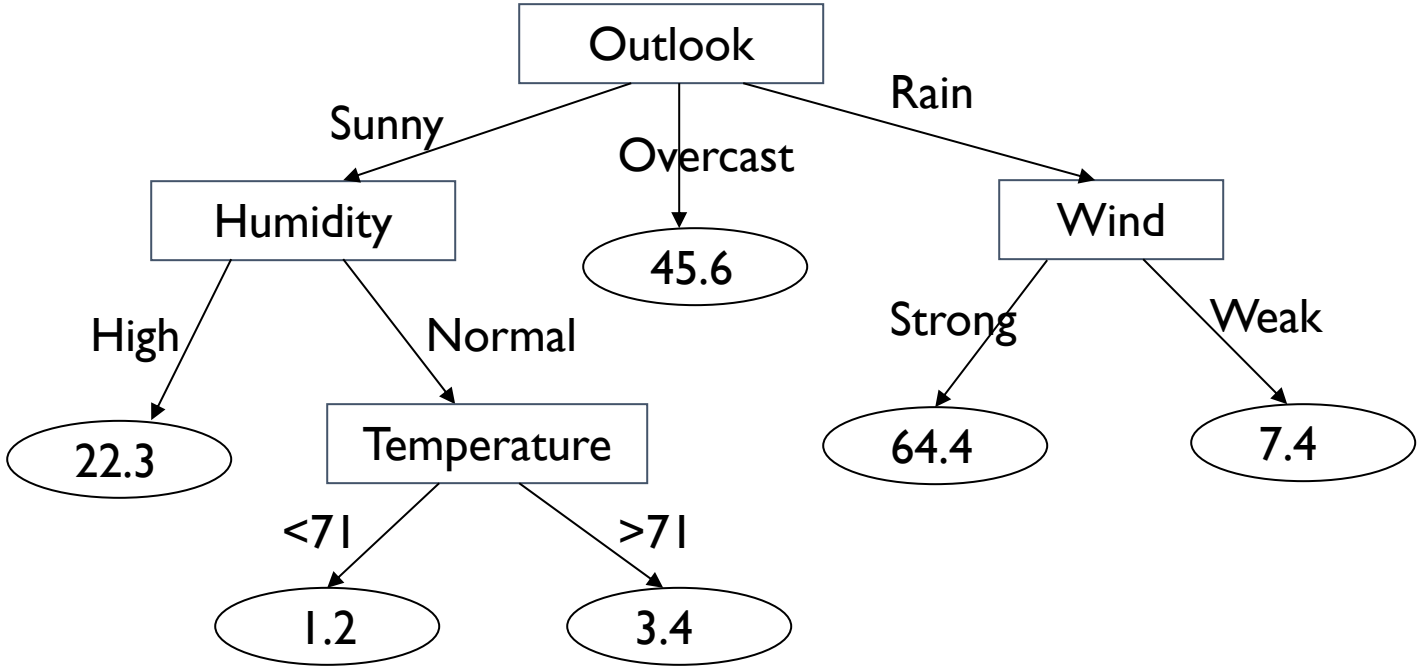
Multivariate Trees





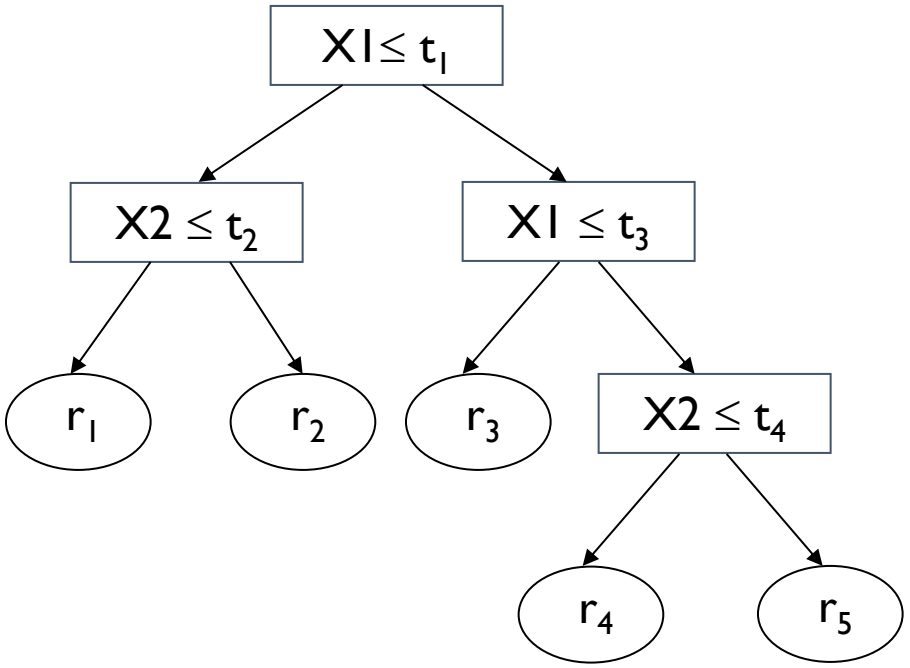
Regression Trees

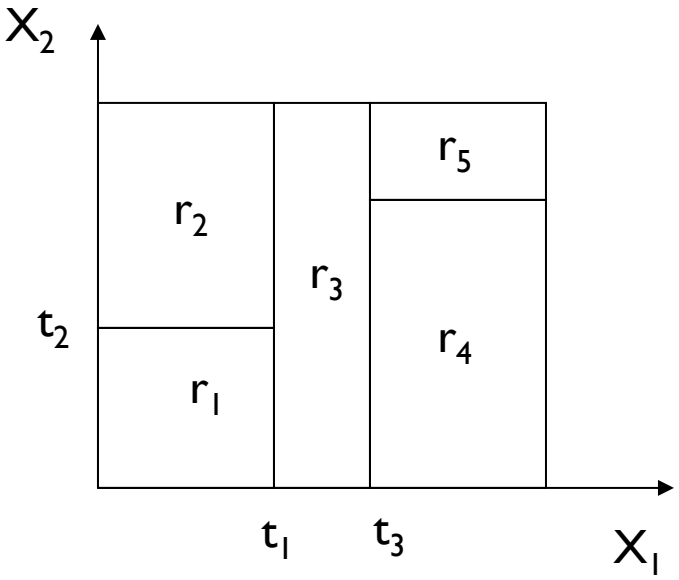
- Tree for regression: exactly the same model but with a number in each leaf instead of a class



Regression Trees

- A regression tree is a piecewise constant function of the input attributes





Growing Regression Trees

- To minimize the square error on the learning sample, the prediction at a leaf is the average output of the learning cases reaching that leaf
- Impurity of a sample is defined by the variance of the output in that sample:

$$I(LS)=var_{y|LS}\{y\}=E_{y|LS}\{(y-E_{y|LS}\{y\})^2\}$$

- The best split is the one that reduces the most variance:

$$\Delta I(LS,A) = \mathrm{var}_{y|LS}\{y\} - \sum_a \frac{|LS_a|}{|LS|}\mathrm{var}_{y|LS_a}\{y\}$$

Regression Tree Pruning

- Exactly the same algorithms apply: pre-pruning and post-pruning.
- In practice, pruning is more important in regression because full trees are much more complex
  - Each data instance can have a different output value and hence the full tree has as many leaves as there are training instances

Readings

- Introduction to Machine Learning, Ethem Alpaydin, Chapter 9 (2nd Edn)
- PRML, Chris Bishop, Chapter 14 (Sec 14. 4), 2006 Edn