Big Data

VICTORIA UNIVERSITY OF WELLINGTON
TE HERENGA WAKA

AIML427

Filter and Embedded Feature Selection

Dr Bach Hoai Nguyen

Bach.Nguyen@vuw.ac.nz

Outline

- Filter feature selection methods (cont.)
- Embedded feature selection methods
- Feature selection applications

Mutual Information

- Mutual information evaluates the information shared between each pair of features/variables

- Relevance:
  - Classification performance
  - The relevance (MI) between each selected feature and the class labels

- Redundancy:
  - Number of features
  - The redundancy (MI) between the selected features

Ranking using Information Theory Measures

- Categorical (nominal) data:
  - If it is a numeric feature it must first be *discretised*

- Mutual information estimation method can used

- Mutual information between a feature and the class labels
  - Rank features
  - Select top ranked features

Filter Method

Objective Function:

$$Rel = \sum_{x_i \in X} I(x_i; C)$$

$$Red = \sum_{\substack{x_i, x_j \in X, \\ and\ i \neq j}} I(x_i; x_j)$$

- X is the selected feature subset
- x_i, x_j : feature in X
- C is the class lables
- *Rel*: relevance between X and c
- *Red*: redundancy within X

$$\begin{aligned} \mathrm{I(X; Y)} &= \mathrm{H(X) - H(X|Y) = H(Y) - H(Y|X)} \\ &= \sum_{x \epsilon X, y \epsilon Y} p(x,y) log_2 \frac{p(x,y)}{p(x)p(y)} \end{aligned}$$

Minimum Redundancy-Maximum Relevance

(mRMR)

- S is the feature subset, Ω is the pool of all candidate features, the minimum redundancy condition is:

$$\min_{s\subset\Omega} \frac{1}{|S|^2} \sum_{i,j\,\in S} I(f_i, f_j)$$

where |S| is the number of features in S.

- For classes $c=(c_i,....c_k)$ the maximum relevance condition maximises the total relevance of all features in *S:*

$$\max_{s\subset\Omega} \frac{1}{|S|} \sum_{i\in S} I(c, f_i)$$

H.C. Peng, F.H. Long, and C. Ding, Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, 2005, pp. 1226–1238.

Minimum Redundancy-Maximum Relevance (mRMR)

- The mRMR feature set optimises these two conditions simultaneously, either in quotient form:

$$\max_{s\subset\Omega}\left\{\frac{\sum_i I(c,f_i)}{\frac{1}{|S|}\sum_{i,j\in S} I(f_i,f_j)}\right\}$$

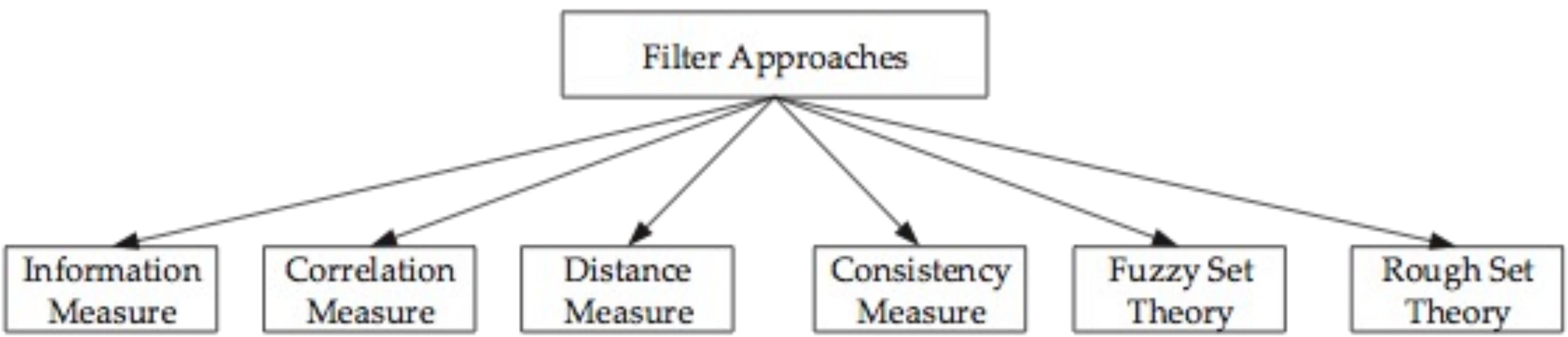
or in difference form:

$$\max_{s\subset\Omega}\left\{\sum_i I(c,f_i) - \frac{1}{|S|}\sum_{i,j\in S} I(f_i,f_j)\right\}$$

H.C. Peng, F.H. Long, and C. Ding, Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, 2005, pp. 1226–1238.

Filter Feature Selection

- Information theory-based approach:
  - max-relevance, and min-redundancy
- Rough set theory for feature selection
- Fast correlation based filter feature selection
- Evolutionary computation for filter feature selection
- …
- Issues:
  - Most filter approaches do not evaluate subsets of features



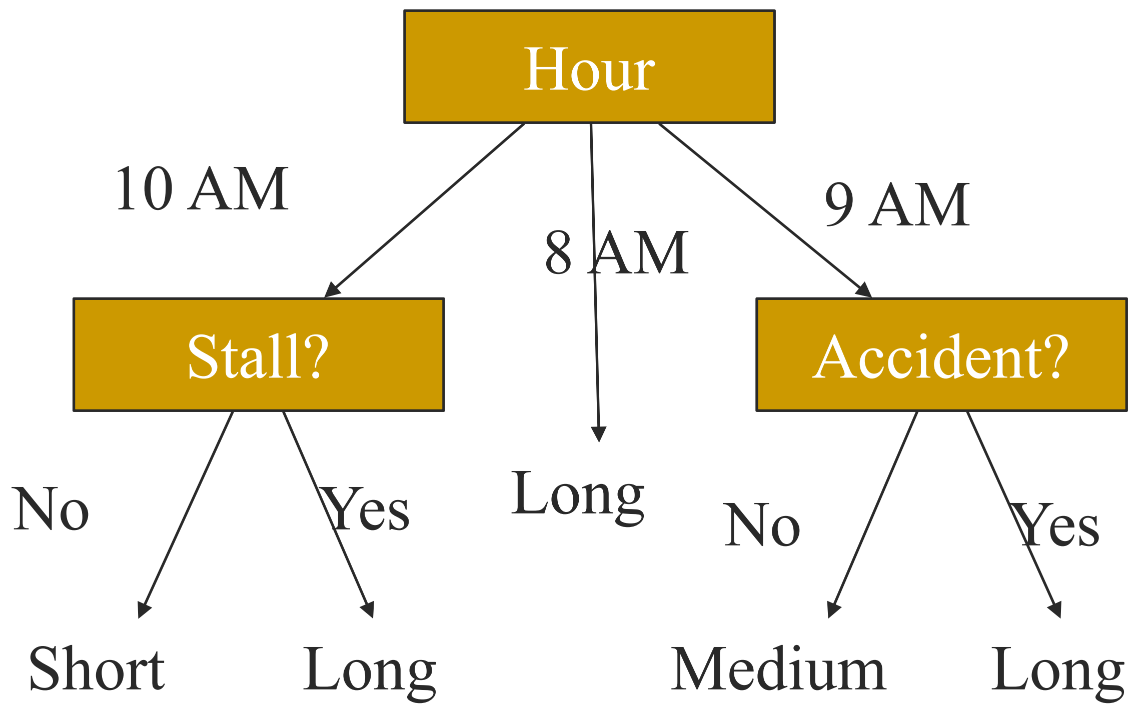
EMBEDDED FEATURE SELECTION

Sample Experience Table / Training Data

Example	**Attributes**				**Target (Class)**
	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

Predicting Commute Time

If we leave at 10 AM and there are no cars stalled on the road, what will our commute time be?



Decision Tree

- In this decision tree, we made a series of Boolean decisions and followed the corresponding branch
  1. Did we leave at 10 AM?
  2. Is there any car stalled on the road?
  3. Is there any accident on the road?

- By answering each of these yes/no questions, we then concluded how long our commute might take

- We do not have to represent this tree graphically
- We could represent it as a set of classification rules – but much harder to read!

Choosing Attributes

- *But the decision tree only showed 3 attributes: hour, accident and stall,* Why is that?

- Methods for selecting attributes show that weather is not a discriminating attribute

- The principle of ***Occam's Razor:*** given a number of competing hypotheses, the simplest one is preferable

- The basic structure of creating a decision tree is the same for most decision tree algorithms

- The difference is in *how* we select the attributes for the tree

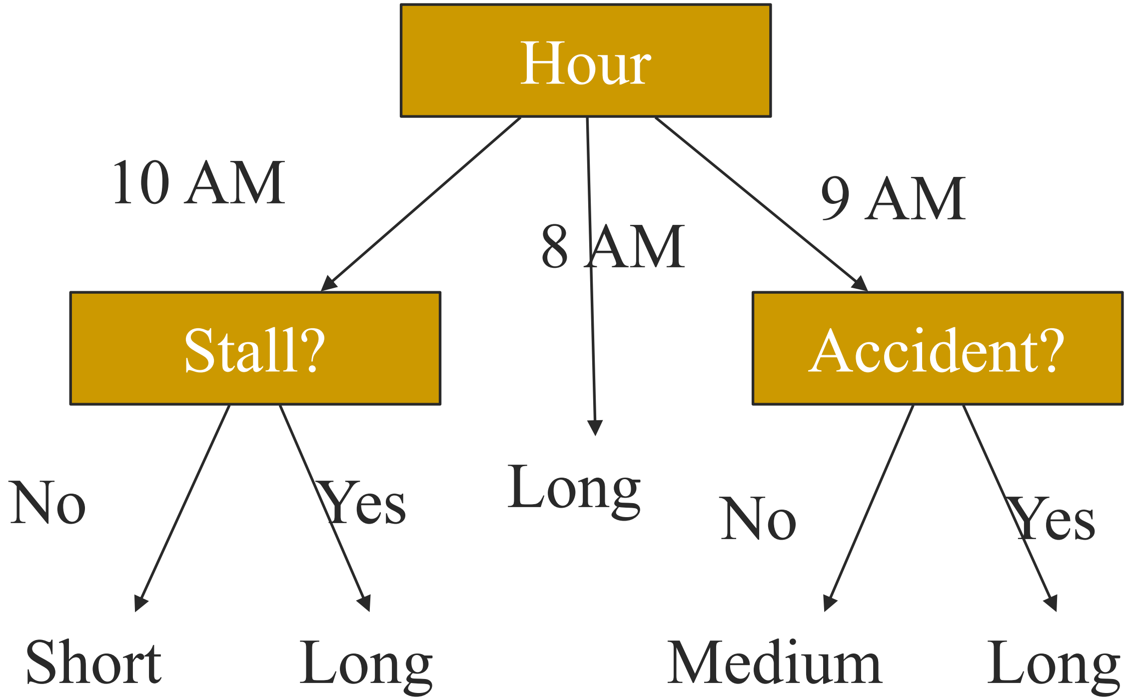
DT Algorithms

The basic idea behind any decision tree algorithm is:

1. Choose the best attribute(s) to split the remaining instances and make that attribute a decision node
2. Repeat this process recursively for each child
3. Stop when:
   - All the instances have the same target attribute value; or
   - There are no more attributes; or
   - There are no more instances

Identifying the Best Attributes

Referring back to our original decision tree:



- How did we know to split on *Hour* and then on *stall* and *accident* and not *weather*?
- Based on the **Entropy** impurity measure

Decision tree versions

- We will focus on the Iterative Dichotomiser 3 (ID3) algorithm developed by Ross Quinlan in 1975
  - ID3 follows the principle of Occam's razor in attempting to create the smallest decision tree possible
- Quinlan expanded the principles of ID3 to create C4.5, C5.0
  - C4.5 improved: discrete and continuous attributes, missing attribute values, attributes with differing costs, pruning trees
  - C5.0: speed/memory improvement, support boosting
    - Commercialised…kind of: https://www.rulequest.com/licensing.html
- KNIME implements C4.5

Pruning: Prepruning and Postpruning

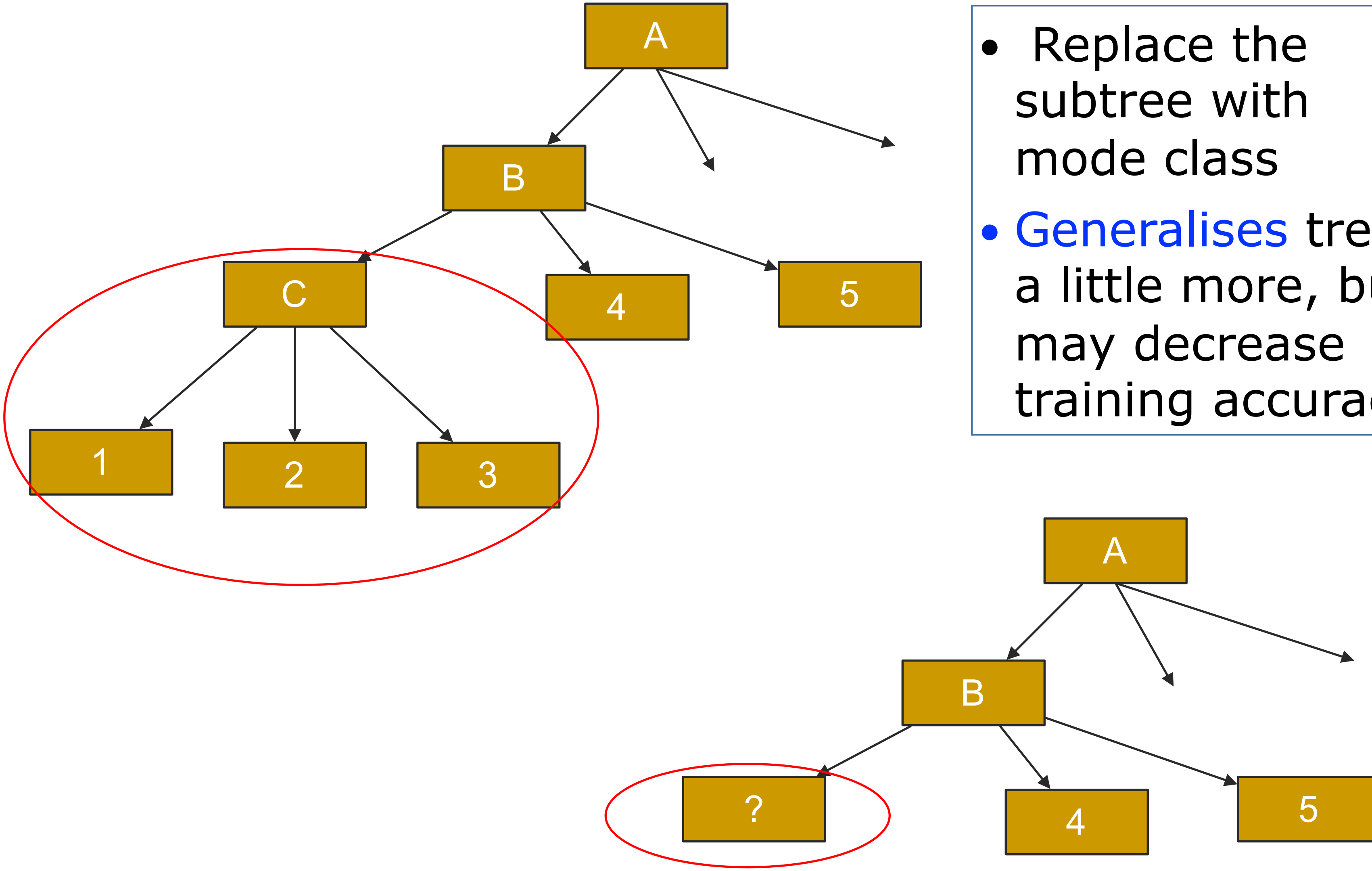
- There is another technique for reducing the number of attributes used in a tree – pruning

- *Prepruning:* decide during the building process when to stop adding attributes (e.g. based on their information gain)
- However, this may be problematic – Why?
  - *Feature interaction*: individual attributes may not contribute much to a decision, but when combined, they may have a significant impact

- *Postpruning:* waits until the full decision tree has built and then prunes the attributes
  - Two techniques: Subtree Replacement and Subtree Raising

Subtree Replacement

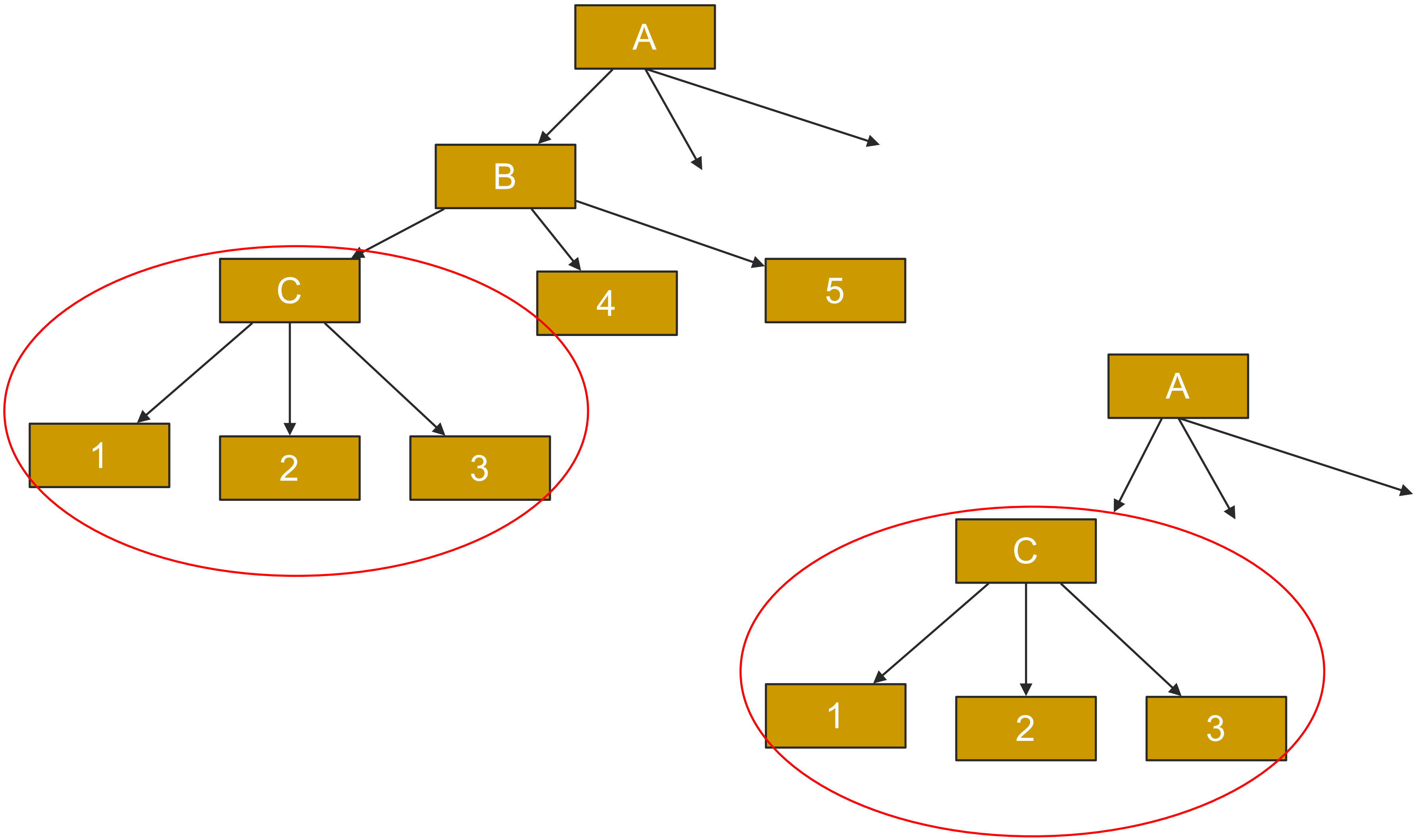
- Entire subtree is replaced by a single leaf node



- Replace the subtree with mode class
- Generalises tree a little more, but may decrease training accuracy

Subtree Raising

- Entire subtree is raised onto another node



Decision Tree

- Decision trees can be used to help predict future results
- The trees are (potentially) easy to understand
- Decision trees work more efficiently with discrete attributes
- Decision tress can deal with missing data

How does a decision tree achieve feature selection?

Problems with DT

- Discretisation method
  - choose cut points (e.g. 9AM) for splitting **continuous** attributes
  - cut points generally lie in a subset of boundary points: two adjacent instances in a sorted list have different class labels
  - Entropy Based Discretisation

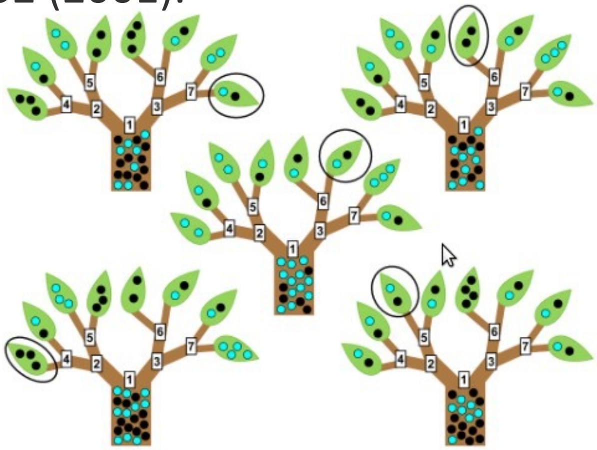
- DTs suffer from errors propagating throughout a tree
  - A very serious problem as the number of classes increases
  - Since DTs work by a series of local decisions, what happens when one of these local decisions is wrong? (*Greedy*)
    - Every decision from that point on may be wrong
    - We may never return to the correct path of the tree

Random forest (RF)

- Random forest (RF) is an ensemble classifier that consists of many decision trees. It predicts the class that is the mode of the predictions by individual trees.

- Extension: “Random Forests”™. Combines Breiman's "bagging" idea and the random selection of features.
  – “Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest.”

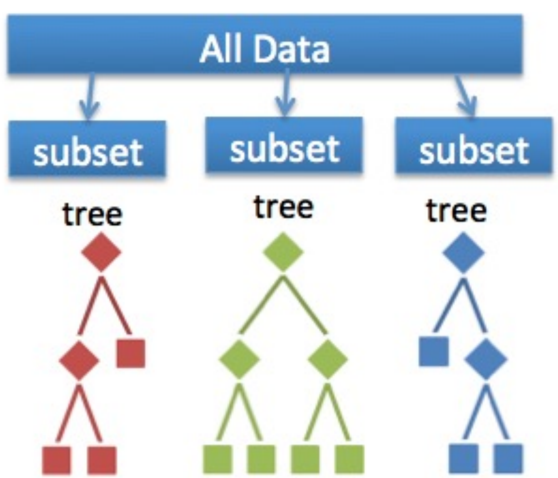
    Breiman, L. Random Forests. *Machine Learning* **45,** 5–32 (2001).
    https://doi.org/10.1023/A:1010933404324

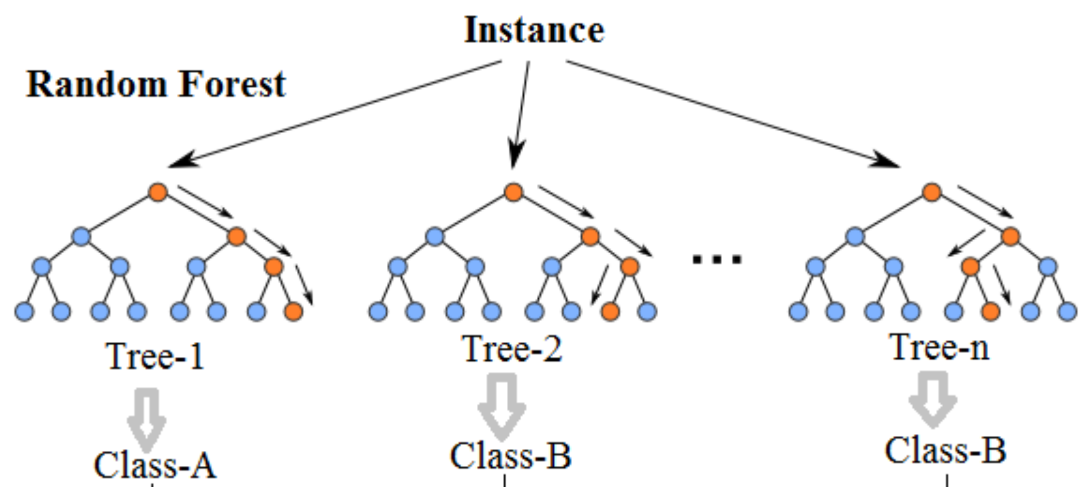


Random forests (RF)

- Voting mechanisms: growing an ensemble of trees and letting them vote for the most popular class.
  - Further improvements in classification accuracy.

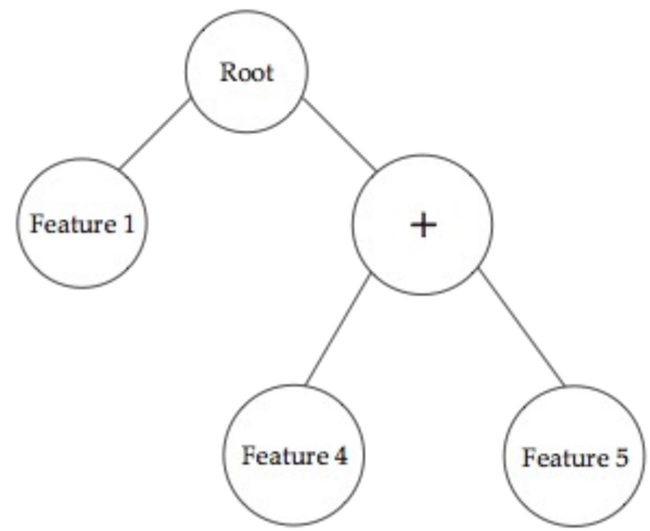
- To grow these ensembles, often random vectors/examples are generated that govern the growth of each tree.





Other Embedded Feature Selection Methods

- Decision trees
- Neural networks
- Support vector machines
- Sparse Logistic Regression
- Probabilistic/Bayesian classifiers
- Genetic programming (GP) (AIML426 in T2)
  - During the evolutionary training process:
    - a GP program as a classifier is learnt
    - a set of features are selected



Feature Selection Applications

- Biological and biomedical tasks
  - gene analysis, biomarker detection, cancer classification, and disease diagnosis
- Image and signal processing
  - image analysis, face recognition, human action recognition, EEG brain-computer-interface, speaker recognition, handwritten digit recognition, personal identification, and music instrument recognition.
- Network/web service
  - Web service composition and development, network security, and email spam detection.

Feature Selection Applications

- Business and financial problems
  - Financial crisis, credit card issuing in bank systems
  - customer churn prediction.

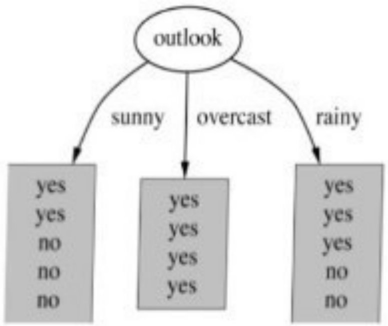
- Others
  - power system optimisation,
  - weed recognition in agriculture,
  - melting point prediction in chemistry,
  - weather prediction.

Example Papers for Reading

- M. Dash and H. Liu, “Feature selection for classification,” Intelligent Data Analysis, vol. 1, no. 4, pp. 131–156, 1997.
- Kohavi, Ron, and George H. John. "Wrappers for feature subset selection." Artificial intelligence 97.1-2 (1997): 273-324.
- I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” The Journal of Machine Learning Research, vol. 3, pp. 1157– 1182, 2003.
- H. Liu, H. Motoda, R. Setiono, and Z. Zhao, “Feature selection: An ever evolving frontier in data mining,” in Feature Selection for Data Mining, vol. 10 of JMLR Proceedings, pp. 4–13, JMLR.org, 2010.
- H. Liu and L. Yu, “Toward integrating feature selection algorithms for classification and clustering,” IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 4, pp. 491–502, 2005.
- Zhai, Yiteng, Yew-Soon Ong, and Ivor W. Tsang. "The Emerging" Big Dimensionality"." *IEEE Computational Intelligence Magazine* 9.3 (2014): 14-26.
- Bing Xue, Mengjie Zhang, Will Browne, Xin Yao. “A Survey on Evolutionary Computation Approaches to Feature Selection”, IEEE Transaction on Evolutionary Computation, vol. 20, no. 4, pp. 606-626, Aug. 2016.
- Bing Xue, Mengjie Zhang and Will Browne. "A Comprehensive Comparison on Feature Selection Approaches to Classification". International Journal of Computational Intelligence and Applications (IJCIA). Vol. 14, No. 2. 2015. pp. 1550008-1 -- 1550008-23.
- Bing Xue, Mengjie Zhang, Will Browne. "Particle swarm optimization for feature selection in classification: A multi-objective approach", IEEE Transactions on Cybernetics, vol. 43, no. 6, pp. 1656-1671, 2013

Calculate entropy

- Play golf? Yes - No



E (Outlook=sunny) = $-\frac{2}{5}\log\left(\frac{2}{5}\right)-\frac{3}{5}\log\left(\frac{3}{5}\right)=0.971$

E (Outlook=overcast) = $-1\log(1)-0\log(0)=0$

E (Outlook=rainy) = $-\frac{3}{5}\log\left(\frac{3}{5}\right)-\frac{2}{5}\log\left(\frac{2}{5}\right)=0.971$

H(S,Outlook)

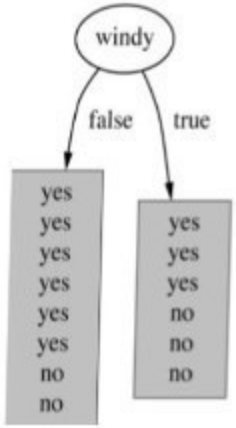
Average Entropy information for Outlook

I (Outlook) = $\frac{5}{14}*0.971+\frac{4}{14}*0+\frac{5}{14}*0.971=0.693$

$$\sum_{t\in T} p(t)H(t)$$

Gain (Outlook) = E(S) – I (outlook) = 0.94-.693 = 0.247

$$IG(A,S)=H(S)-\sum_{t\in T} p(t)H(t)$$



E (Windy=false) = $-\frac{6}{8}\log\left(\frac{6}{8}\right)-\frac{2}{8}\log\left(\frac{2}{8}\right)=0.811$

E (Windy=true) = $-\frac{3}{6}\log\left(\frac{3}{6}\right)-\frac{3}{6}\log\left(\frac{3}{6}\right)=1$

Average entropy information for Windy

I (Windy) = $\frac{8}{14}*0.811+\frac{6}{14}*1=0.892$

Gain (Windy) = E(S) – *I* (Windy) = 0.94-0.892=0.048