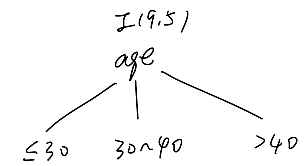
Decision Tree

impurity measurement : Gini Index, Entropy, Misclassification error

① $\text{Entropy}(t) = -\sum_j p(j|t)\log_2 p(j|t)$

Information Gain : $\text{Gain}_{split} = \text{Entropy}(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n}\text{Entropy}(i)\right)$
(spliting based on)



eg.

age	Yes	no	I(yes, no)
≤30	2	3	I(2,3)
30~40	4	0	I(4,0)
>40	3	2	I(3,2)

$I(9,5) = 0.94$

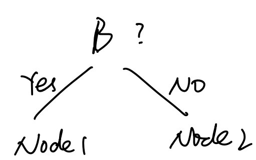
$E(\text{age}) = \frac{5}{14}I(2,3) + \frac{4}{14}I(4,0) + \frac{5}{14}I(3,2)$

$= 0.694$

$\text{Gain}(\text{age}) = I(9,5) - E(\text{age}) = 0.246$

② GINI Index : $\text{GINI}(t) = 1 - \sum_j [p(j|t)]^2$

(spliting based on) $\text{GINI}_{split} = \sum_{i=1}^{k} \frac{n_i}{n}\text{GINI}(i)$



eg.

	N_1	N_2
C_1	5	1
C_2	2	4

parent
C_1 6
C_2 6

$\text{Gini}(N_1) = 1 - (\frac{5}{7})^2 - (\frac{2}{7})^2 = 0.408$

$\text{Gini}(N_2) = 1 - (\frac{1}{5})^2 - (\frac{4}{5})^2 = 0.32$

$\text{Gini}(\text{Children}) = \frac{7}{12} \times 0.408 + \frac{5}{12} \times 0.32 = 0.3713$

$\text{Gini}(\text{parent}) = \frac{1}{2} \times (1 - \frac{1}{4} - \frac{1}{4}) + \frac{1}{2}(1 - \frac{1}{4} - \frac{1}{4}) = \frac{1}{2}$

$\text{Gain} = \frac{1}{2} - 0.3713 = 0.1287$

③ classification error (at node t) : $\text{Error}(t) = 1 - \max_i P(i|t)$

eg.

C_1	0
C_2	6
C_1	1
C_2	5

$P(C_1) = 0/6 = 0$, $P(C_2) = 6/6 = 1$

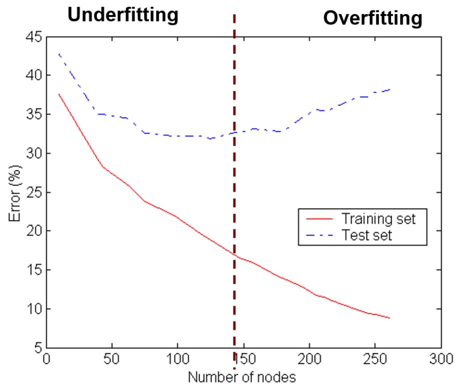
$\text{Error} = 1 - \max(0, 1) = 0$

$P(C_1) = 1/6$, $P(C_2) = 5/6$

$\text{Error} = 1 - \max(1/6, 5/6) = 1/6$

Underfitting and Overfitting

Training and test error



Underfitting: when model is too simple, both training and test errors are large

- Pre-Pruning (Early Stopping Rule)
  - Stop the algorithm before it becomes a fully-grown tree
  - Typical stopping conditions for a node:
    - Stop if all instances belong to the same class
    - Stop if all the attribute values are the same
  - More restrictive conditions:
    - Stop if number of instances is less than some user-specified threshold
    - Stop if class distribution of instances are independent of the available features (e.g., using χ^2 test)
    - Stop expanding the current node when the observed gain in purity measure falls below a certain threshold. (e.g., Gini or information gain).

Avoid Overfitting in Decision Tree

- Overfitting: An induced tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - Poor accuracy for unseen samples
- Two approaches to avoid overfitting
  - Prepruning: Halt tree construction
  - Postpruning: Remove branches from a "fully grown" tree.
    - Use a set of data different from the training data to decide which is the "best pruned tree"

- Post-pruning
  - Grow decision tree to its entirety
  - Trim the nodes of the decision tree in a bottom-up fashion
  - If **generalization error** improves after trimming, replace sub-tree by a leaf node.
  - Class label of leaf node is determined from majority class of instances in the sub-tree

Estimating Generalization Errors

- Methods for estimating generalization errors:
  - Pessimistic approach:

For a decision tree T, Let N_t be the number of training records and $e(T)$ be the number of misclassified records. Ω is the penalty term associated with each leaf node. The pessimistic error estimate, $e_g(T)$, can be computed as follows:

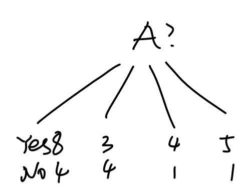
$$e_g(T) = \frac{e(T) + \Omega(T)}{N_t}$$

For example, suppose the penalty term is equal to 0.5

- For each leaf node: e'(t) = (e(t)+0.5)
- Total errors: e'(T) = e(T) + N × 0.5 (N: number of leaf nodes)
- For a tree with 30 leaf nodes and 10 errors on training (out of 1000 instances):
  Training error = 10/1000 = 1%
  Generalization error = (10 + 30×0.5)/1000 = 2.5%

example.

$\boxed{\begin{array}{ll} \text{class = Yes} & 20 \\ \text{class = No} & 10 \\ \text{Error} = 10/30 & \end{array}}$



Training error (Before splitting) = 10/30

Pessimistic error = (10+0.5)/30 = 10.5/30

Training error (After splitting) = 9/30

Pessimistic error (After splitting) = (9+4×0.5)/30 = 11/30

⇒ Prune!

Evaluation of the Classifier

Predictive accuracy, speed and scalability, Robustness, Interpretability.

Method for Performance Evaluation

Holdout, Random Sampling, Cross Validation (k-fold, leave-one-out)

Performance Evaluation…

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

- Most widely-used metric:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Performance Evaluation

$$Sensitivty = \frac{TP}{TP+FN},$$ (True positive rate)

$$Specificity = \frac{TN}{TN+FP}$$ (True negative rate)

$$\text{Precision} = \frac{TP}{TP+FP}$$

The percentage of instances predicted as *"Yes"* that actually are *"Yes"* instance.

Decision Tree: Advantages

- Applicability: no any prior assumption
- Model explains its reasoning -- builds rules
- Build model quickly and extremely fast at classifying unknown records
- No problems with missing data
- Works fine with many dimensions

Decision Tree: disadvantages

- Model has high order interactions -- all splits are dependent on previous splits
- Data are split at each node, making further splits able to use less and less data
- Decision Tree built is typically locally optimal and not globally optimal or best.
- The greedy characteristic of decision trees leads to over-sensitivity to the training set, to irrelevant attributes and to noise make decision trees especially ***unstable***: a minor change in one split close to the root will change the whole subtree below.