Decision Tree Induction: Algorithm

- Basic algorithm
  - Tree is constructed in a **top-down, recursive, divide-and-conquer manner**
  - At start, all the training examples are at the root
  - Examples are partitioned recursively based on selected attributes
  - On each node, attributes are selected based on the training examples on that node, and a heuristic or statistical measure (e.g., information gain)

เลือก root node ที่ดีที่สุด

- Conditions for stopping partitioning
  - All samples for a given node belong to the same class
  - There are no remaining attributes for further partitioning
  - There are no samples left

ถ้าไม่มี Attributes ก็จะไม่ทำต่อ

- Prediction
  - **Majority voting** is employed for classifying the leaf

โดย ใช้กลุ่มที่ไหน เยอะสุด เอาอันนั้น

How to Handle Continuous-Valued Attributes?

- Method 1: Discretize continuous values and treat them as categorical values
  - E.g., age: < 20, 20..30, 30..40, 40..50, > 50

ข้อมูล=ตัวเลข → ใช้การแบ่งกลุ่ม.

- Method 2: Determine the ***best split point*** for continuous-valued attribute A
  - Sort the value A in increasing order:, e.g. 15, 18, 21, 22, 24, 25, 29, 31, …
  - *Possible split point:* the midpoint between *each pair of adjacent values*
    - $(a_i+a_{i+1})/2$ is the midpoint between the values of a_i and a_{i+1}
    - e.g., (15+18/2 = 16.5, 19.5, 21.5, 23, 24.5, 27, 30, …
  - The point with the *maximum information gain* for A is selected as the **split-point** for A
- Split: Based on split point P
  - The set of tuples in D satisfying A ≤ P vs. those with A > P

Gain Ratio: A Refined Measure for Attribute Selection

- Information gain measure is biased towards attributes with a large number of values
- Gain ratio: Overcomes the problem (as a normalization to information gain)

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

  - GainRatio(A) = Gain(A)/SplitInfo(A)
  - The attribute with the maximum gain ratio is selected as the splitting attribute
  - Gain ratio is used in a popular algorithm C4.5 (a successor of ID3) by R. Quinlan
- Example
  - $\mathrm{SplitInfo}_{\mathrm{income}}(\mathrm{D}) = -\frac{4}{14}\log_2\frac{4}{14} - \frac{6}{14}\log_2\frac{6}{14} - \frac{4}{14}\log_2\frac{4}{14} = 1.557$
  - GainRatio(income) = 0.029/1.557 = 0.019

Another Measure: Gini Index

- Gini index: Used in CART, and also in IBM IntelligentMiner
- If a data set D contains examples from n classes, gini index, $gini(D)$ is defined as
  - $gini(D) = 1 - \sum_{j=1}^{n} p_j^2$ ✓

ใหม่ $1 - \left\{\left(\frac{3}{8}\right)^2 + \left(\frac{5}{8}\right)^2\right\}$

    - p_j is the relative frequency of class j in D
- If a data set D is split on A into two subsets D_1 and D_2, the $gini$ index $gini(D)$ is defined as

ตัวเลือก

  - $gini_A(D) = \frac{|D_1|}{|D|}gini(D_1) + \frac{|D_2|}{|D|}gini(D_2)$ ✓

สูตรเดิม

$\frac{8\,(3,5)}{\bigcirc} = \boxed{-\frac{3}{8}\log\frac{3}{8} - \frac{5}{8}\log\frac{5}{8}}$

- Reduction in Impurity:
  - $\Delta gini(A) = gini(D) - gini_A(D)$
- The attribute provides the smallest $gini_{split}(D)$ (or the largest reduction in impurity) is chosen to split the node (*need to enumerate all the possible splitting points for each attribute*)

Method 1 : Categorize

15, 18, 21, 22,

15-16, >18, >22

method2 : best split point

Y Y N Y N N N N

15, 18, 21, 22, 24, 25, 29, 31

แบ่งออกดู

8 ตัว

age

≤ 21.5 / > 21.5

3 ตัว — YYN

5 ตัว — YNNNN

$$Age_{21.5} = \frac{3}{8} I(2,1) + \frac{5}{8} I(1,4)$$

8 ตัว

≤ 24.5 / > 24.5

5 ตัว — YYNYN

3 ตัว — NNN

$$\frac{5}{8} I(3,2) + \frac{3}{8} I(0,3)$$

Method3 : Random

Computation of Gini Index

- Example: D has 9 tuples in buys_computer = "yes" and 5 in "no"

$$gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

- Suppose the attribute income partitions D into 10 in D_1: {low, medium} and 4 in D_2
  - $gini_{income\in\{low,medium\}}(D) = \frac{10}{14} gini(D_1) + \frac{4}{14} gini(D_2)$

$$= \frac{10}{14}\left(1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right) = 0.443$$

$$= Gini_{income\in\{high\}}(D)$$

  - $Gini_{\{low,high\}}$ is 0.458; $Gini_{\{medium,high\}}$ is 0.450
  - Thus, split on the {low,medium} (and {high}) since it has the lowest Gini index
- All attributes are assumed continuous-valued
- May need other tools, e.g., clustering, to get the possible split values
- Can be modified for categorical attributes

Comparing Three Attribute Selection Measures

- The three measures, in general, return good results but
  - **Information gain**:
    - biased towards multivalued attributes
  - **Gain ratio**:
    - 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

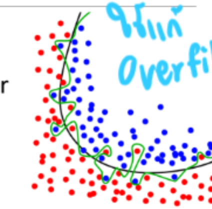
Other Attribute Selection Measures

- Minimal Description Length (MDL) principle
  - Philosophy: The simplest solution is preferred
  - The best tree as the one that requires the fewest # of bits to both (1) encode the tree, and (2) encode the exceptions to the tree
- CHAID: a popular decision tree algorithm, measure based on χ^2 test for independence
- Multivariate splits (partition based on multiple variable combinations)
  - CART: finds multivariate splits based on a linear combination of attributes
- There are many other measures proposed in research and applications
  - E.g., G-statistics, C-SEP
- Which attribute selection measure is the best?
  - Most give good results, none is significantly superior than others

ต้องการสอน/ฝึก.

Overfitting and Tree Pruning

→ ตัดแต่งกิ่ง. ใช้แก้ Overfitting

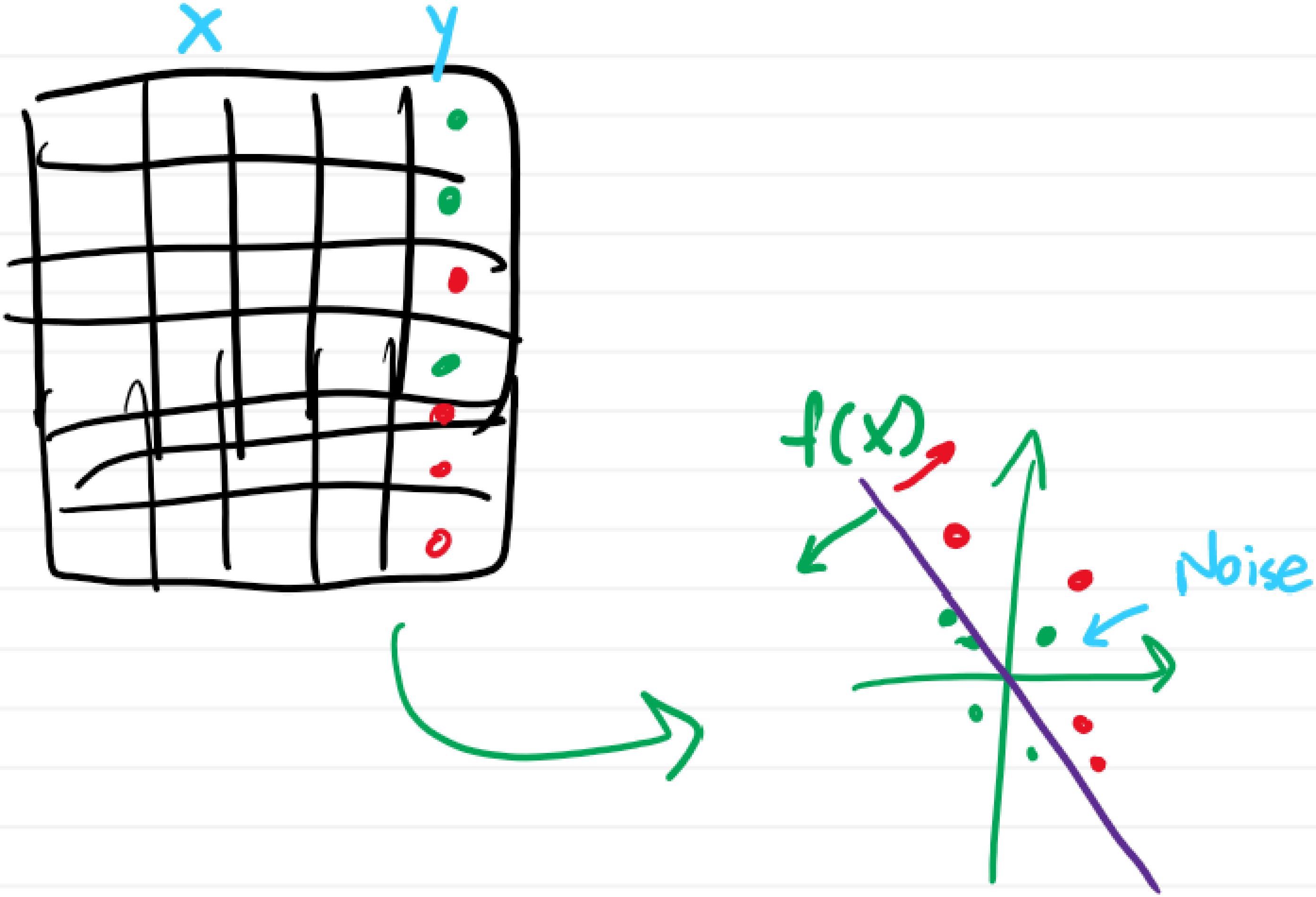


- Overfitting: An induced tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers

    → กิ่งมากเกินไป
  - Poor accuracy for unseen samples
- Two approaches to avoid overfitting

  หยุดการเติบโต → ตามที่กำหนด.
  - Prepruning: *Halt tree construction early*-do not split a node if this would result in the goodness measure falling below a threshold
    - Difficult to choose an appropriate threshold

  ต้นไม้โตเต็มที่ กำหนดกฎเกณฑ์ → ในการตัดกิ่ง.
  - Postpruning: *Remove branches* from a "fully grown" tree—get a sequence of progressively pruned trees
    - Use a set of data different from the training data to decide which is the "best pruned tree"



occum's Razor

Criterion = กำหนดเกณฑ์

splitter = การแยกnode

max_depth = กำหนดชั้นของ decision tree

min_samples_split = จำนวน data ที่น้อยที่สุด
ที่จะให้dataไหลลงมา

min_samples_leaf = เกณฑ์ขั้นต่ำ ของ node

max_leaf_node = กำหนดการแตกของใบ.

max_features = กำหนด features ไว้