Supervised Learning

- Task:
  - Learn (to imitate) a function $f: X \rightarrow Y$
- Training Examples:
  - Learning algorithm is given the correct value of the function for particular inputs $\rightarrow$ training examples
  - An example is a pair $(x, f(x))$, where $x$ is the input and $f(x)$ is the output of the function applied to $x$.
- Goal:
  - Find a function $h: X \rightarrow Y$ that approximates $f: X \rightarrow Y$ as well as possible.

Inductive Learning Strategy

- Strategy and hope (for now, later theory):
  - Any hypothesis $h$ found to approximate the target function $f$ well over a sufficiently large set of training examples $S$ will also approximate the target function well over other unobserved examples.
- Can compute:
  - A hypothesis $h \in H$ such that $h(x) = f(x)$ for all $x \in S$.
- Ultimate Goal:
  - A hypothesis $h \in H$ such that $h(x) = f(x)$ for all $x \in X$.

Version Space

Definition: The version space, $V_{S,H}$, with respect to hypothesis space $H$ and training examples $S$, is the subset of hypotheses from $H$ consistent with all training examples in $S$.

$$V_{S,H} = \{h \in H | h \text{ consistent}(S, H)\}$$
List-Then-Eliminate Algorithm

- Init VS ← H
- For each training example \((x, y) \in S\)
  - remove from VS any hypothesis \(h\) for which \(h(x) \neq y\)
- Output VS

Top-Down Induction of DT (simplified)

Training Data: \(S = \{(x_1, y_1), \ldots, (x_n, y_n)\}\)

\[\text{TIDDT}(S, y_{\text{def}})\]

- IF (all examples in \(S\) have same \(y\))
  - Return leaf with class \(y\) (or class \(y_{\text{def}}, \) if \(S\) is empty)
- ELSE
  - Pick \(A\) as the “best” decision attribute for next node
  - FOR each value \(v_i\) of \(A\) create a new descendent of node
    - \(S_i = \{(x, y) \in S : \text{attribute } A \text{ of } x \text{ has value } v_i\}\)
    - Subtree \(t_i\) for \(v_i\) is \(\text{TIDDT}(S_i, y_{\text{def}})\)
  - RETURN tree with \(A\) as root and \(t_i\) as subtrees

Which Attribute is “Best”? 

\[\begin{array}{c|c|c|c}
29, 35 & 21, 5 & 8, 30 & 18, 33 & 11, 2
\end{array}\]

Example: TDIDT

\[\text{TIDDT}(S, y_{\text{def}})\]

- IF (all ex in \(S\) have same \(y\))
  - Return leaf with class \(y\) (or class \(y_{\text{def}}, \) if \(S\) is empty)
- ELSE
  - Pick \(A\) as the “best” decision attribute for next node
  - FOR each value \(v_i\) of \(A\) create a new descendent of node
    - \(S_i = \{(x, y) \in S : \text{attribute } A \text{ of } x \text{ has value } v_i\}\)
    - Subtree \(t_i\) for \(v_i\) is \(\text{TIDDT}(S_i, y_{\text{def}})\)
  - RETURN tree with \(A\) as root and \(t_i\) as subtrees

Example: Text Classification

- Task: Learn rule that classifies Reuters Business News
  - Class +: “Corporate Acquisitions”
  - Class -: Other articles
  - 2000 training instances
- Representation:
  - Boolean attributes, indicating presence of a keyword in article
  - 9947 such keywords (more accurately, word “stems”)
Decision Tree for “Corporate Acq.”

Learned tree:
- has 437 nodes
- is consistent

Accuracy of learned tree:
- 11% error rate

Note: word stems expanded for improved readability.