Supervised Function Approximation

- Problem setting
  - Set of possible instances $X$
  - Unknown target function $f: X \rightarrow Y$
  - Set of function hypotheses: $H = \{h \mid h: X \rightarrow Y\}$

- Input
  - Training examples $\{<X(i), Y(i)>\}$ of unknown function $f$

- Output
  - Hypothesis $h \in H$ that best approximates $f$
Supervised Function Approximation: **Decision Tree Learning**

- **Problem setting**
  - Set of possible instances $X$
    - each instance $x$ in $X$ is a feature vector $x = <x_1, x_2, \ldots, x_n>$
  - Unknown target function $f: X \rightarrow Y$
    - $Y$ is discrete valued
  - Set of function hypotheses: $H = \{h \mid h: X \rightarrow Y\}$
    - each hypothesis $h$ is a decision tree

- **Input**
  - Training examples $\{<X^{(i)}, Y^{(i)}>|\}$ of unknown function $f$

- **Output**
  - Hypothesis $h \in H$ that best approximates $f$
Searching for the best hypothesis

- Decision tree learning
  - performs a heuristic search through space of decision trees
The big picture
No free lunch!

- Inductive inference
  - Reliable generalization beyond the training data is impossible unless we add more assumptions into the model.

- “Essentially all models are wrong, but some are useful.”
  — George Box
No free lunch!

- Inductive inference
  - Reliable generalization beyond the training data is impossible unless we add more assumptions into the model.

- “Essentially all models are wrong, but some are useful.”
  — George Box

Q. What was the assumption in decision tree learning?
Which tree should we output?

- Decision tree learning
  - performs a heuristic search through space of decision trees
Assumption in decision tree learning

- Stop the top-down greedy growth of decision tree at smallest acceptable tree. Why?
  - Prefer the simplest hypothesis that fits the data (Occam’s Razor)
Assumptions (or the lack of it) have implications...

- What if we let the decision tree learning algorithm to grow freely at will?
  - This may lead to overfitting!
Overfitting in decision tree learning

![Graph showing overfitting in decision tree learning](image)
Avoiding overfitting

- How to avoid overfitting?
  - Stop growing the tree when data split is not significant
  - Grow full tree, then post-prune

- How to select the “best” tree?
  - Measure performance on training dataset
  - Measure perforce on standalone validation dataset
Reduce-error pruning

Split data into *training* and *validation* set

Create tree that classifies *training* set correctly

Do until further pruning is harmful:

1. Evaluate impact on *validation* set of pruning each possible node (plus those below it)

2. Greedily remove the one that most improves *validation* set accuracy

**Produces smallest version of most accurate subtree**