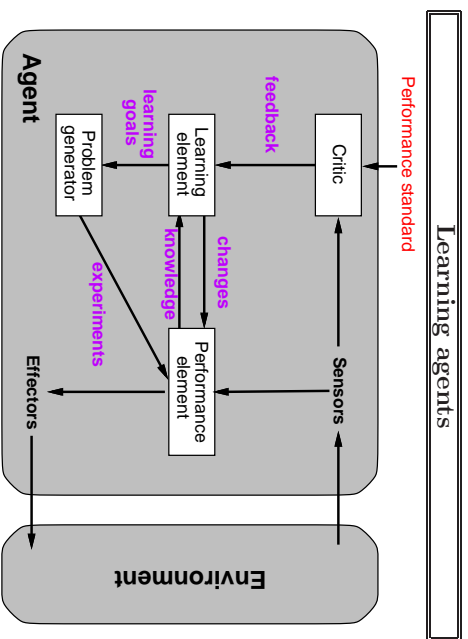


LEARNING FROM OBSERVATIONS

CHAPTER 18, SECTIONS 1–3



Chapter 18, Sections 1-3 1

Chapter 18, Sections 1-3 4

Outline

- ◇ Learning agents
- ◇ Inductive learning
- ◇ Decision tree learning
- ◇ Measuring learning performance

Learning element

Design of learning element is dictated by

- ◇ what type of performance element is used
- ◇ which functional component is to be learned
- ◇ how that functional component is represented
- ◇ what kind of feedback is available

Example scenarios:

Performance element	Component	Representation	Feedback
Alpha-beta search	Eval. fn.	Weighted linear function	Win/loss
Logical agent	Transition model	Successor-state axioms	Outcome
Utility-based agent	Transition model	Dynamic Bayes net	Outcome
Simple reflex agent	Percept-action fn	Neural net	Correct action

Supervised learning: correct answers for each instance

Reinforcement learning: occasional rewards

Chapter 18, Sections 1-3 2

Chapter 18, Sections 1-3 3

Learning

Learning is essential for unknown environments,
i.e., when designer lacks omniscience

Learning is useful as a system construction method,

i.e., expose the agent to reality rather than trying to write it down

Learning modifies the agent's decision mechanisms to improve performance

Inductive learning (a.k.a. Science)

Simplest form: learn a function from examples (*tabula rasa*)

f is the target function

An example is a pair $x_i, f(x_i)$, e.g.,

$$\frac{0|0|X}{X|X|}, +1$$

Problem: find a(n) hypothesis h

such that $h \approx f$
given a training set of examples

(This is a highly simplified model of real learning:

- Ignores prior knowledge
- Assumes a deterministic, observable “environment”
- Assumes examples are given
- Assumes that the agent wants to learn f —why?)

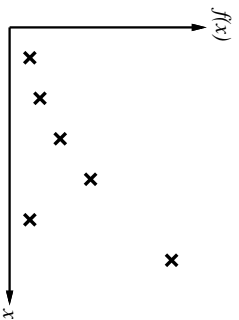
Chapter 18, Sections 1-3 3

Chapter 18, Sections 1-3 6

Inductive learning method

Construct/adjust h to agree with f on training set
(h is consistent if it agrees with f on all examples)

E.g., curve fitting:

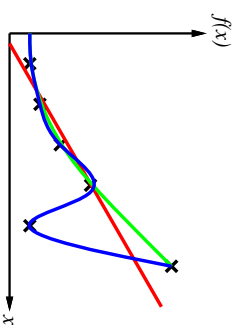


Chapter 18, Section 1-3 7

Inductive learning method

Construct/adjust h to agree with f on training set
(h is consistent if it agrees with f on all examples)

E.g., curve fitting:

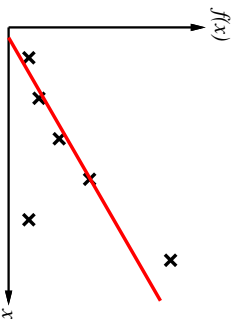


Chapter 18, Section 1-3 10

Inductive learning method

Construct/adjust h to agree with f on training set
(h is consistent if it agrees with f on all examples)

E.g., curve fitting:

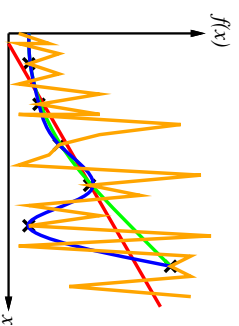


Chapter 18, Section 1-3 8

Inductive learning method

Construct/adjust h to agree with f on training set
(h is consistent if it agrees with f on all examples)

E.g., curve fitting:

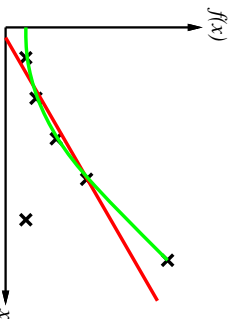


Chapter 18, Section 1-3 11

Inductive learning method

Construct/adjust h to agree with f on training set
(h is consistent if it agrees with f on all examples)

E.g., curve fitting:

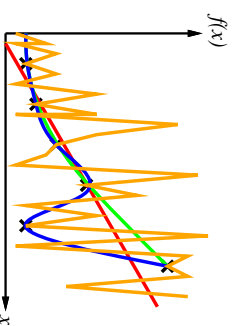


Chapter 18, Section 1-3 9

Inductive learning method

Construct/adjust h to agree with f on training set
(h is consistent if it agrees with f on all examples)

E.g., curve fitting:



Chapter 18, Section 1-3 12

Ockham's razor: maximize a combination of consistency and simplicity

Attribute-based representations

Examples described by attribute values (Boolean, discrete, continuous, etc.)
 E.g., situations where I will/won't wait for a table:

Example	Attributes							Target			
	All	Bar	Fri	Hun	Pat	Price	Rain		Res	Type	Est
X ₁	T	F	F	T	Some	\$\$\$	F	F	French	0-10	T
X ₂	T	F	F	T	Full	\$	F	F	Thai	30-60	F
X ₃	F	T	F	F	Some	\$	F	F	Burger	0-10	T
X ₄	T	F	F	T	Full	\$	F	F	Thai	10-30	T
X ₅	T	F	T	F	Full	\$\$\$	F	T	French	>60	F
X ₆	F	T	F	F	Some	\$	T	T	Italian	0-10	T
X ₇	F	T	F	F	None	\$	T	F	Burger	0-10	F
X ₈	F	F	T	F	Some	\$	T	T	Burger	>60	F
X ₉	F	F	T	F	Full	\$	T	F	Thai	0-10	T
X ₁₀	T	T	T	T	Full	\$\$\$	F	F	Italian	10-30	F
X ₁₁	F	F	F	F	None	\$	F	F	Thai	0-10	F
X ₁₂	T	T	F	F	Full	\$	F	F	Burger	30-60	T

Classification of examples is positive (T) or negative (F)

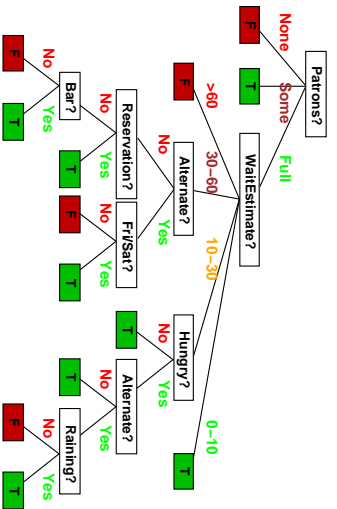
Hypothesis spaces

How many distinct decision trees with n Boolean attributes??

Decision trees

One possible representation for hypotheses

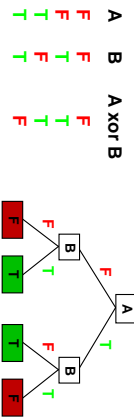
E.g., here is the "true" tree for deciding whether to wait:



Expressiveness

Decision trees can express any function of the input attributes.

E.g., for Boolean functions, truth table row \rightarrow path to leaf:



Trivially, there is a consistent decision tree for any training set w/ one path to leaf for each example (unless f nondeterministic in x) but it probably won't generalize to new examples
 Prefer to find more **compact** decision trees

Hypothesis spaces

How many distinct decision trees with n Boolean attributes??

= number of Boolean functions

Hypothesis spaces

How many distinct decision trees with n Boolean attributes??

= number of Boolean functions

= number of distinct truth tables with 2^n rows

Hypothesis spaces

How many distinct decision trees with n Boolean attributes??

- = number of Boolean functions
- = number of distinct truth tables with 2^n rows = 2^{2^n}

Chapter 16, Section 1-3 19

Hypothesis spaces

How many distinct decision trees with n Boolean attributes??

- = number of Boolean functions
- = number of distinct truth tables with 2^n rows = 2^{2^n}

Chapter 16, Section 1-3 22

Hypothesis spaces

How many distinct decision trees with n Boolean attributes??

- = number of Boolean functions
- = number of distinct truth tables with 2^n rows = 2^{2^n}

E.g., with 6 Boolean attributes, there are 18,446,744,073,709,551,616 trees

Chapter 16, Section 1-3 20

Hypothesis spaces

How many distinct decision trees with n Boolean attributes??

- = number of Boolean functions
- = number of distinct truth tables with 2^n rows = 2^{2^n}

E.g., with 6 Boolean attributes, there are 18,446,744,073,709,551,616 trees

How many purely conjunctive hypotheses (e.g., $Hungry \wedge \neg Rain$)??

Chapter 16, Section 1-3 21

- Each attribute can be in (positive), in (negative), or out
 - $\Rightarrow 3^n$ distinct conjunctive hypotheses
- More expressive hypothesis space
 - increases chance that target function can be expressed 😊
 - increases number of hypotheses consistent w/ training set
 - \Rightarrow may get worse predictions 😞

How many purely conjunctive hypotheses (e.g., $Hungry \wedge \neg Rain$)??

E.g., with 6 Boolean attributes, there are 18,446,744,073,709,551,616 trees

Decision tree learning

Aim: find a small tree consistent with the training examples

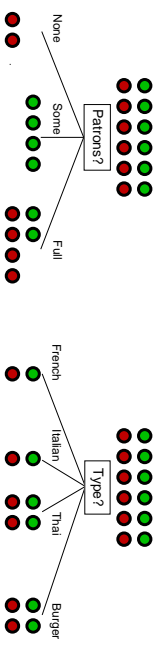
Idea: (recursively) choose “most significant” attribute as root of (sub)tree

```
function DTL(examples, attributes, default) returns a decision tree
  if examples is empty then return default
  else if all examples have the same classification then return the classification
  else if attributes is empty then return MODE(examples)
  else
    best ← CHOOSE-ATTRIBUTE(attributes, examples)
    tree ← a new decision tree with root test best
    for each value  $v_i$  of best do
      examples $i$  ← {elements of examples with best =  $v_i$ }
      subtree ← DTL(examples $i$ , attributes – best, MODE(examples))
      add a branch to tree with label  $v_i$  and subtree subtree
    return tree
```

Chapter 16, Section 1-3 23

Choosing an attribute

Idea: a good attribute splits the examples into subsets that are (ideally) “all positive” or “all negative”



Patrons? is a better choice—gives **information** about the classification

Chapter 16, Section 1-3 24

Information

Information answers questions

The more clueless I am about the answer initially, the more information is contained in the answer

Scale: 1 bit = answer to Boolean question with prior (0.5, 0.5)

Information in an answer when prior is $\langle P_1, \dots, P_n \rangle$ is

$$H(P_1, \dots, P_n) = \sum_{i=1}^n -P_i \log_2 P_i$$

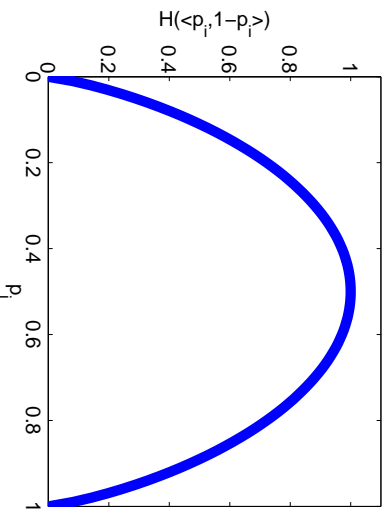
(also called entropy of the prior)

Waiting for tables

Example	Attributes											Target
	All	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait	
X ₁	T	F	F	T	Some	\$\$\$	F	T	French	0-10	T	
X ₂	T	F	F	T	Full	\$	F	F	Thai	30-60	F	
X ₃	F	T	F	F	Some	\$	F	F	Burger	0-10	T	
X ₄	F	T	T	T	Full	\$	F	F	Thai	10-30	T	
X ₅	T	F	T	F	Full	\$\$\$	F	T	French	>60	F	
X ₆	F	T	F	F	Some	\$	T	T	Burger	0-10	T	
X ₇	F	F	F	F	None	\$	T	F	Burger	0-10	F	
X ₈	F	F	T	T	Some	\$	T	T	Thai	>60	F	
X ₉	F	F	T	T	Full	\$	T	F	Burger	0-10	T	
X ₁₀	F	T	T	T	Full	\$\$\$	F	F	Italian	10-30	F	
X ₁₁	F	F	F	F	None	\$\$\$	F	F	Thai	0-10	F	
X ₁₂	T	F	T	T	Full	\$	F	F	Burger	30-60	T	

$$\sum_{i=1}^{12} \frac{p_i + n_i}{p + n} H\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right)$$

Information contd.



Information contd.

Suppose we have p positive and n negative examples at the root

⇒ $H((p/(p+n), n/(p+n)))$ bits needed to classify a new example
E.g., for 12 restaurant examples, $p = n = 6$ so we need 1 bit

An attribute splits the examples E into subsets E_i , each of which (we hope) needs less information to complete the classification

Let E_i have p_i positive and n_i negative examples

⇒ $H((p_i/(p_i+n_i), n_i/(p_i+n_i)))$ bits needed to classify a new example
⇒ **expected** number of bits per example over all branches is

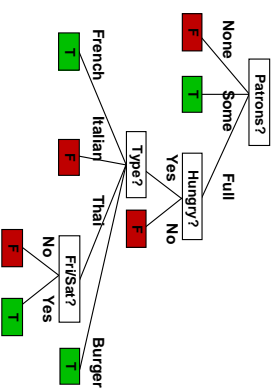
$$\sum_i \frac{p_i + n_i}{p + n} H\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right)$$

For *Patrons?*, this is 0.459 bits, for *Type?* this is (still) 1 bit

⇒ choose the attribute that minimizes the remaining information needed

Example contd.

Decision tree learned from the 12 examples:



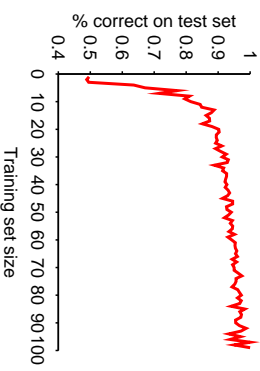
Substantially simpler than “true” tree—a more complex hypothesis isn’t justified by small amount of data

Performance measurement

How do we know that $h \approx f$? (Hume’s **Problem of Induction**)

- 1) Use theorems of computational/statistical learning theory
- 2) Try h on a new test set of examples
(use **same distribution** over **example space** as training set)

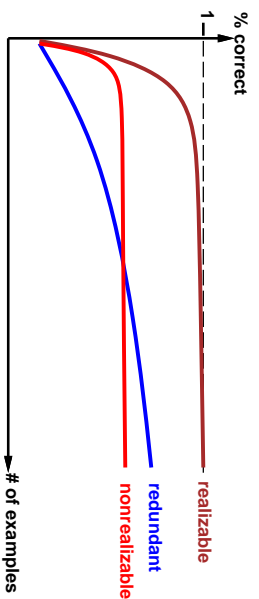
Learning curve = % correct on test set as a function of training set size



Performance measurement contd.

Learning curve depends on

- **realizable** (can express target function) vs. **non-realizable**
- non-realizability can be due to missing attributes
- or restricted hypothesis class (e.g., thresholded linear function)
- redundant expressiveness (e.g., loads of irrelevant attributes)



Chapter 16, Section 1-3 31

Summary

Learning needed for unknown environments, lazy designers

Learning agent = performance element + learning element

Learning method depends on type of performance element, available feedback, type of component to be improved, and its representation

For supervised learning, the aim is to find a simple hypothesis that is approximately consistent with training examples

Decision tree learning using information gain

Learning performance = prediction accuracy measured on test set

Chapter 16, Section 1-3 32