Foundations of Artificial Intelligence

CS472/3

Lecture #29

Bart Selman

Slide CS472–1

- Introduction to Learning

Slide CS472–2

From previous time:

• expert systems example of **knowledge-intensive** AI.

• 90's: shift to **compute-intensive**
data-intensive AI (e.g., statistical)
& **reasoning / infererence intensive** methods
(e.g., diagnosis from first principles (NASA)).

Slide CS472–3

Concludes Knowledge Repr. and Reasoning

Declarative representations:
general representations about the world
combined with reasoning procedures.
Propositional and first-order logic:
syntax and semantics
Key: soundness and completeness —
enables purely **symbolic** reasoning.
Specialized reasoners: e.g. expert systems
Part III, R&N (pp. 151 – 334)

Slide CS472–4

Avoiding the **Knowl. Acquisition Bottleneck**

Learning!

Part VI or R&N. pp 523 – 648.
Also, **adaptation / dynamic**.

Can be an humbling experience.

SRI: mid 1960's — Recognizing tanks ....

(McCarthy decided not to study learning!)

CMU: mid 1980's — Off-road driving —
watch for those trees!

Field is turning around

- Learning backgammon (1992)
- Speech recognitions (1990+)
- datamining:
  credit card approval (AMEX / neural net)
  scientific data (image data indexing / NASA)
  DNA sequencing

Brief Glance at the Issues

Q. What can we learn from examples?
Learn programs? (avoid having to program)

E.g., examples of lists together with their sorted versions
Could we learn sorting algorithm?
What's the difficulty?
E.g., examples of numbers each labeled "prime" / "not prime"
Could we learn primality testing algorithm?

E.g., examples of programs each labeled ”halts” / ”doesn’t halt”

Could we learn halting algorithm?

What’s the difficulty?

E.g., randomly played chess games, labels ”win”/”loss”/”draw”

Could we learn chess playing algorithm?

Slide CS472–9

E.g., millions of documents on WWW.

Could we learn NLU program?

E.g., examples of actions in the Wumpus world?

Could we learn rules about the Wumpus world?

Slide CS472–10

Two immediate issues:

- **raw size** of search space
- fundamental questions concerning **induction** and **generalization**

Size of search space

Assume, 10 Binary input properties /features/ bits (up to 1024 distinct numbers).

One output bit: yes/no

How many distinct Boolean functions?

2^{2^n}, i.e., $2^{2^{10}} = 2^{1024} \approx 10^{300}$!!

often have hundreds of input features...

```
input features          output
0 0 0 0 0 0 0 0 0 0 --- 0/1
0 0 0 0 0 0 0 0 0 1 --- 0/1
0 0 0 0 0 0 0 0 1 0 --- 0/1
0 0 0 0 0 0 0 0 1 1 --- 0/1
0 0 0 0 0 0 0 1 0 0 --- 0/1
0 0 0 0 0 0 0 1 0 1 --- 0/1
0 0 0 0 0 0 0 1 1 0 --- 0/1
0 0 0 0 0 0 0 1 1 1 --- 0/1
etc. 1024 entries in table: 2^1024 different tables!
```

Slide CS472–13

Induction

Rote learning: How about simply storing each prime example seen so far?

Problem: Terrible **generalization** ...

Will lead us to Occam's razor:
Prefer the simplest hypothesis that fits the data.
(William of Occam 1320)

E.g., learning concept of "even number".
What's the simplest hypothesis? Compare rote learning.

Slide CS472–14

Key point: all learning can be seen as learning the representation of a function.
Will become clearer with more examples!

Example representations:

- propositional if-then rules
- first-order if-then rules
- first-order logic theories
- decision trees
- neural networks
- Java programs

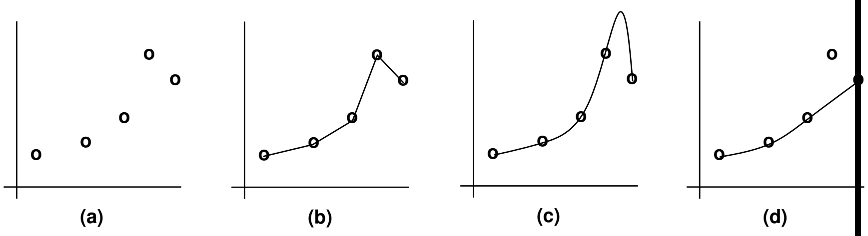
Inductive Learning

Given a collection of examples $(x, f(x))$, return a function h that approximates f.

h is called the **hypothesis** and is chosen from the **hypothesis space**.

What if f is not in hypothesis space?

Induction



Examples could be drawn from (b), (c), or (d).

Slide CS472–17

How does learning algorithm decide?

algorithms have a build in **bias** that

leads them to prefer one hypothesis over another.

Two types of bias:

- **preference bias or search bias**

  depending on how the hypothesis space is explored,
  you get different answers

- **restriction bias or language bias**

  e.g. language: piece-wise linear functions: gives (b)/(d).

Slide CS472–18

Tradeoff (similar in reasoning):
more expressive the language, the harder to find (compute) a good hypothesis
Compare: propositional Horn clauses with first-order logic theories or Java programs.
Also, often need more examples.

Overview

We'll consider:
- Decision trees (chap. 18)
- Neural nets (chap. 19)
- Reinforcement learning (chap 20)
- Knowledge in learning (chap 21)
  - (e.g., physicist looking at bubble-chamber photos)