Analytical Inductive Programming as a Cognitive Rule Acquisition Devise

Ute Schmid, Martin Hofmann, Emanuel Kitzelmann

Cognitive Systems Group
University of Bamberg

AGI 2009

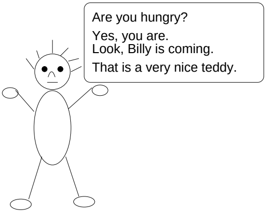
Analytical Inductive Programming

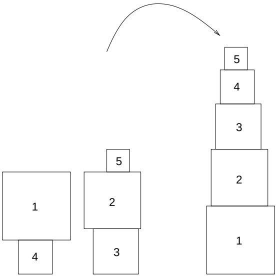
- Inductive Programming (IP): Learning programs from incomplete specifications
  Programs: typically declarative, often functional
  Specifications: typically I/O examples
- **Generate and Test approaches**
- **Analytical approaches:** example-driven
  $\hookrightarrow$ detection of regularities in I/O examples guides generalization
- Intended application: Progamming-Assistance

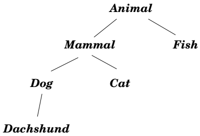
Take a Broader Perspective

Analytical IP provides a mechanism to extract generalized sets of recursive rules from small sets of positive examples of some desired behavior

- Domains where humans are typically exposed to positive examples only:
  - language
  - problem solving traces
  - semantic relations







From LAD to RAD

- Chomsky's claim of a Language Acquisition Device
- Universal mechanism to extract grammar rules from language experience
- LAD is an inductive learning mechanism for recursive rule sets
- Grammar rules characterize linguistic competence and the systematicity, productivity and compositionality of language

- Analytical IP is an inductive learning mechanism for recursive rule sets with language as a special case
  $\hookrightarrow$ Analytical IP is a possible model for a general cognitive rule acquisition device
- We explore this proposition with IGOR2

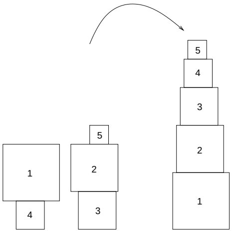
IGOR2 (again)

- Efficient induction of recursive rule sets from small sets of positive examples
- Can learn linear, tree, mutual recursive sets
- Performs necessary function invention
- Can consider background knowledge
- Restriction bias: functional recursive programs where outmost function is either non-recursive or provided by background knowledge
- Non-recursive classifier programs as special case (e.g., `PlayTennis`)
- Preference bias: fewer case distinctions, most specific patterns, fewer recursive calls
- Can be applied to learning generalized rules in various cognitive domains, such as problem solving, reasoning, and natural language processing

Generalized Rules in Problem Solving

The Tower Example

- Even small children learn very fast how to stack blocks in a given sequence
- No "stupid" strategies such as first put all blocks on the table and then stack them in the desired order but optimal strategy
- IGOR2 learns `Tower` from 9 examples of towers with up to four blocks in 1.2 sec



One of the 9 Examples

```
eq Tower(s s table,
    ((s s s s table) (s table) table | ,
      (s s s table) (s s table) table | ,
        nil)) =
  put(s s table, s table,
    put(s s s table, table,
      put(s s s s table, table,
        ((s s s s table) (s table) table | ,
          (s s s table) (s s table) table | ,
            nil))))
```

- Examples are equations with the given state specified in the head and the optimal action sequence (generated by a planner) as body
- additionally: 10 corresponding examples for `Clear` and `IsTower` predicate as background knowledge

Generalized Tower Rule Set

```
Tower(O, S) = S if IsTower(O, S)
Tower(O, S) =
  put(O, Sub1(O, S),
    Clear(O, Clear(Sub1(O, S),
      Tower(Sub1(O, S), S))))   if not(IsTower(O, S))
Sub1(s(O), S) = O .
```

Put the desired block x on the one which has to be below y in a situation where both blocks are clear and the blocks up to the block y are already a tower.

Learning from Positive Experience

RAD in Action

- Solve some problems of a domain using a search-based strategy (e.g. planning), observe regularities in the problem solving traces and generalize over them
- For future problems of this domain: application of the learned rules (no need to search anymore = expertise)

↪ Learning is more than chunking or updating strengthes
↪ Learning as acquisition of new problem solving schemes

Further examples;

- `Clearblock` (4 examples,0.036 sec)
- `Rocket` (3 examples, 0.012 sec)
- `Tower of Hanoi` (3 examples, 0.076 sec)

Learning a Phrase-Structure Grammar

- (Learning rule sets for reasoning, e.g. transitivity of `ancestor`, `isa`)

Learning rules for natural language processing: e.g. a phrase structure grammar

1: *The dog chased the cat.*

2: *The girl thought the dog chased the cat.*

3: *The butler said the girl thought the dog chased the cat.*

4: *The gardener claimed the butler said the girl thought the dog chased the cat.*

$$\begin{aligned} \texttt{S} &\longrightarrow \texttt{NP VP} \\ \texttt{NP} &\longrightarrow \texttt{d n} \\ \texttt{VP} &\longrightarrow \texttt{v NP} \mid \texttt{v S} \end{aligned}$$

Wrapping Up

- Human learning is amazingly powerful
- This power is only partially covered by typical machine learning approaches
- Chunking rules and manipulating rule strengthes (as used in cognitive architectures) is not enough:
  models a cognitive system where all strategic knowledge is already available and only its application needs to be optimized
- Analytical IP provides a mechanism to model acquisition of sets of recursive rules
- Such rules represent problem solving schemes/strategies which are induced from experience