Knowledge Engineering

Semester 2, 2004-05

Michael Rovatsos
mrovatso@inf.ed.ac.uk

Lecture 18 – Knowledge Evolution II: Inductive Logic Programming
15th March 2005

Where are we?

Last time ...

- Knowledge Evolution
- Truth Maintenance Systems (JTMS, ATMS)
- Knowledge in Learning
- Explanation-based Learning

Today ...

- Inductive Logic Programming

Inductive Logic Programming (ILP)

- Rigorous approach to knowledge-based inductive learning problem
- Methods for inducing general, first-order theories from examples
- Using FOL to represent learning hypotheses is useful where attribute-based mathods (e.g. decision trees) fail
- In particular: ILP allows for capturing **relationships between** objects rather than only their attributes
- Hypotheses generated are relatively easy for humans to understand

Today's lecture

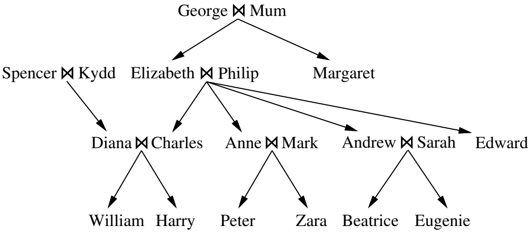
- We will first discuss an extended example
- ... then present a method for top-down ILP
- ... look at inverse induction methods
- and finally discuss the ability of ILP to make discoveries

Example

- Recall entailment constraint of general knowledge-based induction problem:

$$\mathit{Background} \wedge \mathit{Hypothesis} \wedge \mathit{Descriptions} \models \mathit{Classifications}$$

- Example: learning family relationships from examples
- Descriptions given by following family tree:



Example

- Corresponding logical facts:

$$
\begin{array}{ll}
Father(Philip, Charles) & Father(Philip, Anne) \\
Mother(Mum, Margaret) & Mother(Mum, Elizabeth) \\
Married(Diana, Charles) & Married(Elizabeth, Philip) \\
Male(Philip) & Male(Charles) \\
Female(Beatrice) & Female(Margaret) \\
\ldots & \ldots
\end{array}
$$

- Target concept to be learned *Grandparent*, complete set of classifications would be $20 \times 20 = 400$ facts of the form

$$
\begin{array}{ll}
Grandparent(Mum, Charles) & Grandparent(Elizabeth, Beatrice) \\
\neg Grandparent(Mum, Harry) & \neg Grandparent(Spencer, Peter) \\
\ldots & \ldots
\end{array}
$$

Example

- Suppose *Background* is empty
- One possible hypothesis:

$$
\begin{aligned}
Grandparent(x,y) \Leftrightarrow \quad & [\exists z\ Mother(x,z) \wedge Mother(z,y)] \\
\vee \quad & [\exists z\ Mother(x,z) \wedge Father(z,y)] \\
\vee \quad & [\exists z\ Father(x,z) \wedge Mother(z,y)] \\
\vee \quad & [\exists z\ Father(x,z) \wedge Father(z,y)]
\end{aligned}
$$

- What would an attribute-based learning algorithm do here:
  - Turn pairs into objects: $Grandparent(\langle Mum, Charles \rangle)$
  - Descriptions hard to represent,
    e.g. $FirstElementIsMotherOfElizabeth(\langle Mum, Charles \rangle)$
  - Definition of *Grandparent* would become a large disjunction with no generalisation capabilities
- Pincipal advantage of ILP: applicability to relational predicates
  ➡ can cover much wider range of problems

Example

- Additional background knowledge can be used to obtain more concise hypotheses
- Suppose we know
  $Parent(x, y) \Leftrightarrow [Mother(x, y) \lor Father(x, y)]$
- Then we could represent our previous hypothesis as

$$Grandparent(x, y) \Leftrightarrow [\exists z\ Parent(x, z) \land Parent(z, y)]$$

- Even more interesting property of ILP algorithms: **creating** new predicates (e.g. *Parent*)
- **Constructive induction**: one of the hardest problems in machine learning, but some ILP methods can do it!
- We discuss two methods: a generalisation of decision-tree methods & technique based on inverting resolution proofs

FOIL: Top-Down Inductive Learning

- Grow a hypothesis starting from a very general rule, but using a set of first-order clauses rather than a decision tree (clauses used are Horn clauses with negation as failure)
- More specialised clauses are generated by adding conditions to the rule in the following way:
  - Literals can be added using predicates (including goal predicate) with only variables as their arguments
  - Each literal must include at least one variable already appearing in the rule
  - Equality and inequality constraints, arithmetic comparisons
- Large branching factor, but typing information may be used to reduce it
- Heuristic for choice of literal similar to information gain, and hypotheses that are longer than the total length of examples are removed

Example

Example: we are trying to learn the *Grandfather* relation

1. Split examples into positive and negative ones (12/388):
   +: $\langle Mum, Charles \rangle$, $\langle Elizabeth, Beatrice \rangle$
   -: $\langle Mum, Harry \rangle$, $\langle Spencer, Peter \rangle$
2. Construct a set of clauses, each with $Grandfather(x, y)$ as a head
   - Start with $true \Rightarrow Grandfather(x, y)$
   - This classifies negative examples as true, specialise it
   - Generate possible hypotheses by adding a literal to the LHS:
   $$Father(x, y) \Rightarrow Grandfather(x, y)$$
   $$Parent(x, y) \Rightarrow Grandfather(x, y)$$
   $$Father(x, z) \Rightarrow Grandfather(x, y)$$
   - Prefer the one that classifies most data correctly (here: the third one)
3. Repeat these steps until all data is correctly classified

Inductive Learning with Inverse Resolution

- Basic idea: inverting the normal deductive proof process
- Recall resolution rule:

$$\frac{\alpha \vee \beta, \quad \neg\beta \vee \gamma}{\alpha \vee \gamma}$$

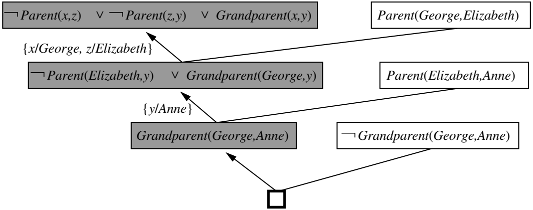
- Resolution is complete, so one must be able to prove

$$\mathit{Background} \wedge \mathit{Hypothesis} \wedge \mathit{Descriptions} \models \mathit{Classifications}$$

- If we can "run the proof backward", we should be able to find *Hypothesis* such that proof succeeds
- Inverse single resolution step takes the resolvent and produces two clauses or the resolvent and one clause and produces one new clause

Example

- Take positive example *Grandparent*(*George*, *Anne*) and start with empty clause, i.e. contradiction and construct the following proof backwards:



  - write $\neg Parent(x,z) \vee \neg Parent(z,y) \vee Grandparent(z,y)$ as $Parent(x,z) \wedge \neg Parent(z,y) \Rightarrow Grandparent(z,y)$
- We have a resolution proof that descriptions, hypothesis and background knowledge entail the classification *Grandparent*(*George*, *Anne*)

Making Discoveries with ILP

- Inverse resolution is a complete algorithm for learning first-order theories (we should always be able to generate hypothesis from examples)
- Could we discover laws of gravity (quantum mechanics, the theory of relativity, etc.)?
  - In theory, yes, but (as with monkey that might write "Hamlet" with a typewriter)
  - We need better heuristics!
- But ILP is able to invent new predicates, and will often do so

Making discoveries with ILP

- For example, for the resolvent

  $$\neg Father(George, y) \vee Ancestor(George, y)$$

  we might generate the two clauses
  - $\neg Father(x, y) \vee P(x, y)$
  - $\neg P(George, y) \vee Ancestor(George, y)$

  in an inverse resolution step (where P is a new predicate)
- A latter step might hypothesize that $Mother(x, y) \Rightarrow P(x, y)$ and $Father(x, y) \Rightarrow P(x, y)$ whereby P would obtain the meaning of *Parent*
- Difficult to predict whether such a new predicate will cover a whole set of observations in a simpler/more elegant way than before

Critique

- Search space in generating new hypotheses can be huge, particularly in inverse induction
  - In particular, anything from descriptions, classifications or background knowledge is a potential candidate
  - Some techniques (e.g. use of linear resolution, restricted representation languages, requiring that all hypothesized clauses be consistent with each other)
- However, it is the most elegant and impressive inductive learning method
  - Simulates human discovery process while making use of prior knowledge

Critique

- Has been successfully used in a number of interesting domains:
  - Solving exercises from standard Prolog textbook
  - Discovery of rules for protein folding
  - Predicting efficacy of drugs from their molecular structures
  - NLP: derive complex relations from text
- When ILP succeeds, its advantage is that the discovered rules can be interpreted by humans

Summary

- Discussed inductive logic programming
- Exceeds the expressiveness of attribute-based inductive learning methods by using FOL representations
- Advantage over other knowledge-based learning methods (e.g. EBL)
  - Not only generalises from existing rules, but may discover new ones altogether!
- Top-down ILP vs. inverse deduction based ILP
  - Trade-off between expressiveness and simplicity
- And with this ...
  - **we have reached the end of this course!**