An approach to Predicate Invention based on Statistical Relational Model

Stefano Ferilli[1,2], Giuseppe Fatiguso[1]

[1]Dipartimento di Informatica

[2]Centro Interdipartimentale per la Logica e sue Applicazioni

Università di Bari

stefano.ferilli@uniba.it

#AI4 AI*IA 2015 CONFERENCE

Ferrara, 22-25 settembre 2015

Predicate Invention

- Branch of symbolic Machine Learning aimed at discovering emerging concepts in the available knowledge
  - The outcome may have important consequences on the efficiency and effectiveness of many kinds of exploitation of the available knowledge
    - Theory restructuring
- Fundamental problems
  - How to handle the combinatorial explosion of candidate concepts to be invented
  - How to select only those that are really relevant

Motivation & Proposal

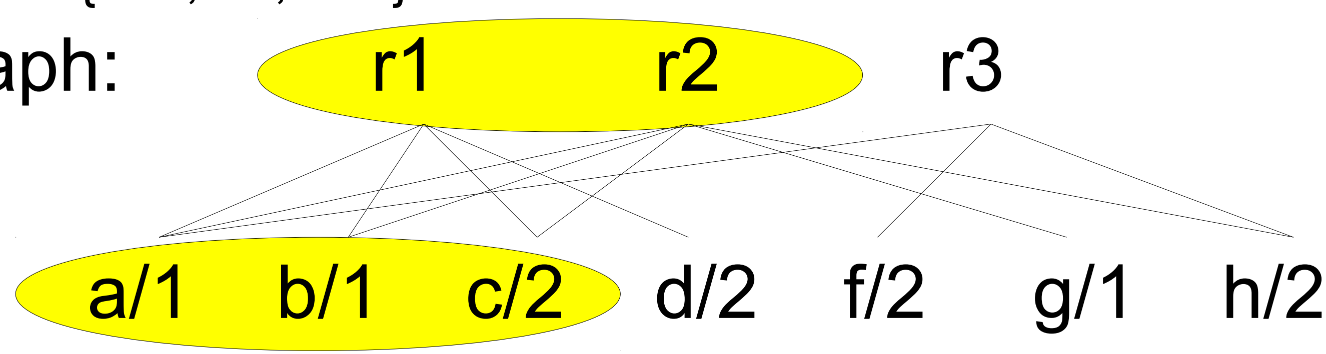
- Complex problem
  - Huge number of candidate concepts
    - Need for automatic techniques to select the best candidates by relevance
      - Purely logical approaches may be too rigid
      - Statistical solutions may provide the required flexibility
        - SPI = Statistical Predicate Invention
  - Indeterminacy in First-Order Logic
- Proposal: Weighted Predicate Invention (WPI)
  - Statistical Relational Learning approach
    - Top-down (Candidate predicates identified in a logic theory, rather than in the background knowledge)
    - Markov Logic Networks (MLN) framework used to assess the relevance of candidate predicate definitions

Search for a pattern

- Define a bipartite graph G
  - Nodes
    - *upper nodes* = rules in the theory
    - *lower nodes* = predicates in the theory
  - Edges: each rule connected to all the predicates appearing in its body
- Among all possible pairs I = (π,ρ)
  - π is a set of lower nodes (made up of at least two elements) that are connected to the same upper-node
  - ρ is the set of rules in the theory that include π.
- Pick one that maximizes (wrt set inclusion) π
  - Predicates appearing in such I's will be used to form a candidate *pattern* to define a predicate to be invented

Search for a Pattern

- Example: Theory R made up of three rules
  - r1 : q(X) :- a(X), b(Y), b(W), c(X,Y), d(Y,W).
    - Predicates: { a/1, b/1, c/2, d/2 }
  - r2 : q(X) :- a(X), b(W), c(X,Y), c(Y,W), g(X), h(Z,Y).
    - Predicates: { a/1, b/1, c/2, g/1, h/2 }
  - r3 : q(X) :- a(X), f(Z,Y), h(X,Y).
    - Predicates: { a/1, f/2, h/2 }
- Bipartite graph:



- Maximal intersection of lower-nodes: I = (π,ρ)
  - π = {a/1, b/1, c/2 }, ρ = { r1, r2 }

Candidate Selection

- For each predicate in π take the minimum number of occurrences across rules in ρ
- Consider all subsets of rules in ρ that follow this pattern (*configurations*) and find a configuration that is present in all rules
  - If no such a configuration exists, remove one occurrence of a predicate and try again
    - Until subsets of two literals are tried
- Build the rule that defines the predicate *i* to be invented
  - Body: the selected configuration
  - Head: the arguments of *i* are the different variables in the selected configuration

Candidate Selection

- Example:
  - Minimum number of literals for all predicates in {a/1,b/1,c/2} is 1
    - Literals for defining an invented predicates: { a(.), b(.), c(.,.) }
  - Configurations:
      - r1: γ_{11} = {a(X),b(Y),c(X,Y)}, γ_{12} = {a(X),b(W),c(X,Y)}
      - r2: γ_{21} = {a(X),b(W),c(X,Y)}, γ_{22} = {a(X),b(W),c(Y,Z)}
    - Best configuration: $\gamma_{12} \equiv \gamma_{21}$
  - Invented rule:
    - i(X,Y,W) :- a(X),b(W),c(X,Y).

Candidate Validation

- Introducing the *invented rule* in the original theory must not decrease the relevance of the existing rules
  - Need of an estimator of the relevance of a rule in the context defined by the given theory and the facts in the background knowledge
    - Weights learned by the MLN weight learning functionality
- Build two MLNs
  - The former simply adds the invented rule to the initial theory
    - Invented predicate is not present in the other rules
    - Invented rule disjoint from the rest of the graph
    - The weights of the other rules do not to change
  - The latter also applies the invented rule to the existing rules
    - The body of some rules in the original theory has changed
    - The invented rule is no more disjoint in the graph
    - Variation of the rule weights expected
  - Invented predicate considered as relevant if the weight in the latter template is greater than the weight in the former

Candidate Validation

- In the previous example, one would get:
  - r0 : i(X,Y,W) :- a(X),b(W),c(X,Y).
  - r1 : q(X) :- b(Y),d(Y,W),i(X,Y,W).
  - r2 : q(X) :- c(Y,W),g(X),h(Z,Y),i(X,Y,W).
  - r3 : q(X) :- a(X),f(Z,Y),h(X,Y).
- Run Discriminative Weight Learning on both templates
  - Two sets of weighted first-order rules
    - $w'_0, w'_1, ..., w'_k$ the weights of rules in the former MLN
    - $w''_0, w''_1, ..., w''_k$ the weights of rules in the latter MLN
  - Invented rule validated if no weight after the application of the invented predicate is less than it was before
    - Otherwise, the invented rule is not added to the theory
  - WPI can be run again on the new theory in order to invent further predicates. Iterating this procedure yields a wider theory restructuring.

Discussion

- Problems
  - Risk of combinatorial explosion for the search space of the groups of literals that define the invented predicate
    - Typical problem of PI
    - Main cause: variable number of literals per predicate for each rule in the pattern
      - More literals per predicate, more possible configurations
  - Cost of evaluating Discriminative Weight Learning twice for every predicate we can invent
- Solution
  - Instead of analyzing this problem from a theoretical or structural viewpoint, we propose an operational model
    - Avoids the invention of trivial or useless concepts

Results

- Effectiveness of predicate invention and theory restructuring
  - WPI applied on theories learned using InTheLEx
- Train Problem (classical) dataset
  - 20 examples of Eastbound or Westbound trains, with the goal to predict Eastbound ones.
- Leave-One-Out Cross-Validation to avoid overfitting
  - Different folds $\rightarrow$ different theories $\rightarrow$ different predicates invented

Experimental results

- Quantitative analysis
  - 4.25 new concepts invented on average in each fold
  - Size of the theories (number of rules) more than doubled on average after invention/restructuring
    - Significantly increases, but with some variability
  - Avg number of literals per rule in the theories dropped from 18.41 to 5.30 on average
    - 28.79% compression ratio
    - Also considering the increase in number of rules
- Qualitative analysis
  - Invention in many folds of the concept that any railway car in the train is somehow connected to the locomotive:
    - car(Car), has_car(Train,Car).