

Lecture 14: Analytical Learning

Cognitive Systems II - Machine Learning

WS 2006/2007

Part III: Learning Programs and Strategies

Explanation-Based Generalization

Motivation

- Using prior knowledge and deductive reasoning to augment information given by trainings examples
- \hookrightarrow explanation-based learning
- Two variants: background-knowledge is or is not complete and correct
- Reach more accuracy with less examples!

Inductive Learning

Given:

- Instance space X
- Hypothesis space H
- Training examples D of some target function f .

$$D = \{\langle x_1, f(x_1) \rangle, \dots, \langle x_n, f(x_n) \rangle\}$$

Determine:

- A hypothesis from H consistent with training examples D .

Analytical Learning

Given:

- Instance space X
- Hypothesis space H
- Training examples D of some target function f .

$$D = \{\langle x_1, f(x_1) \rangle, \dots, \langle x_n, f(x_n) \rangle\}$$

- *Domain theory B for explaining training examples*

Determine:

- A hypothesis from H consistent with both the training examples D and domain theory B .

We say

- B “explains” $\langle x, f(x) \rangle$ if $x + B \vdash f(x)$
- B is “consistent with” h if $B \not\vdash \neg h$

SafeToStack(x,y) Learning Problem

Given:

- Instances: pairs of physical objects
Hypotheses: Sets of Horn clause rules, e.g.,

$$SafeToStack(x, y) \leftarrow Volume(x, vx) \wedge Type(y, Box)$$

- Training Examples: typical example is

SafeToStack(Obj1, Obj2)

On(Obj1, Obj2)

Owner(Obj1, Fred)

Type(Obj1, Box)

Owner(Obj2, Louise)

Type(Obj2, Endtable)

Density(Obj1, 0.3)

Color(Obj1, Red)

Material(Obj1, Cardbd)

- Domain Theory:

SafeToStack(x, y) \leftarrow \neg Fragile(y)

SafeToStack(x, y) \leftarrow Lighter(x, y)

Lighter(x, y) \leftarrow Wt(x, wx) \wedge Wt(y, wy) \wedge Less(wx, wy)

...

SafeToStack(x,y) cont.

Determine:

- A hypothesis from H consistent with training examples and domain theory.

Prolog-EBG

Prolog-EBG(*TargetConcept*, *Examples*, *DomainTheory*)

- *LearnedRules* \leftarrow {}
- *Pos* \leftarrow the positive examples from *Examples*
- for each *PositiveExample* in *Pos* that is not covered by *LearnedRules*, do

1. *Explain:*

- *Explanation* \leftarrow an explanation (proof) in terms of *DomainTheory* that *PositiveExample* satisfies *TargetConcept*

2. *Analyze:*

- *SufficientConditions* \leftarrow the most general set of features of *PositiveExample* that satisfy *TargetConcept* according to *Explanation*.

Prolog-EBG cont.

- (for each *PositiveExample* in *Pos* that is not covered by *LearnedRules*, do)

3. Refine:

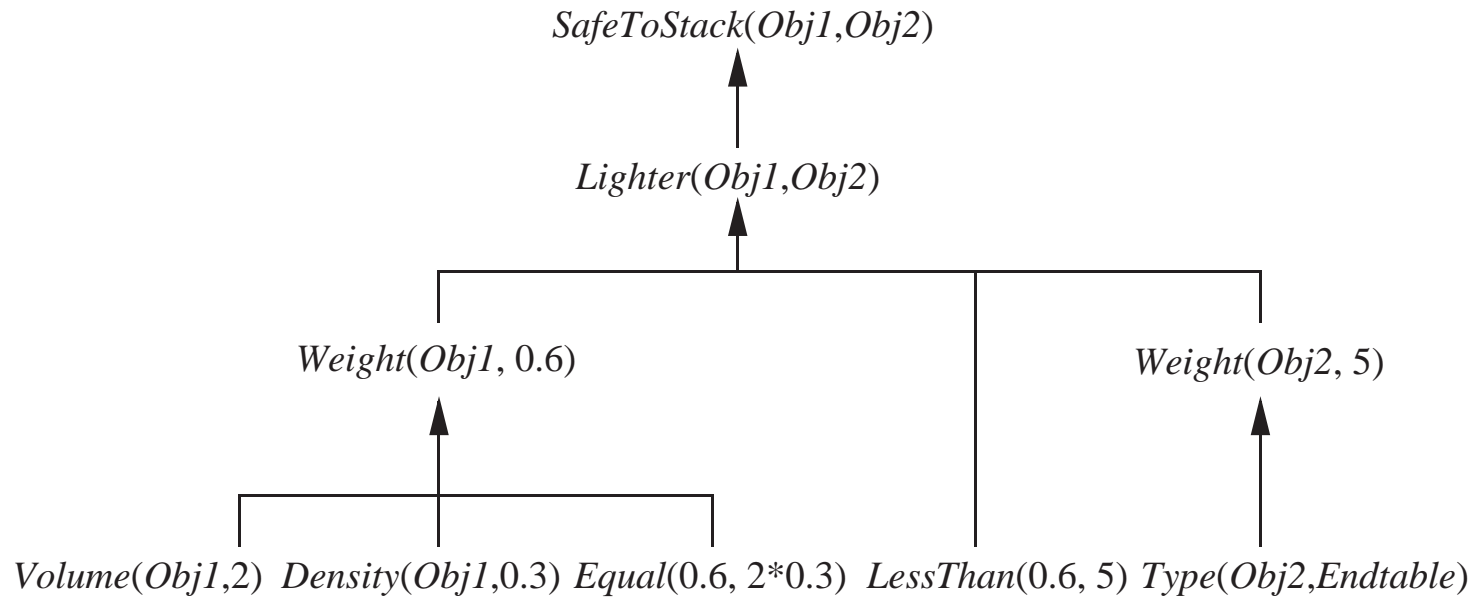
- $LearnedRules \leftarrow LearnedRules + NewHornClause$,
where *NewHornClause* is of the form

$$TargetConcept \leftarrow SufficientConditions$$

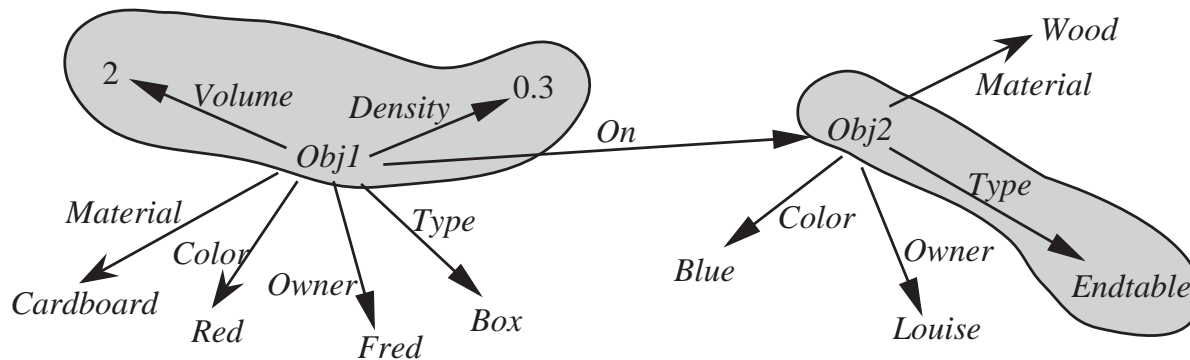
- Return *LearnedRules*

An Explanation

Explanation:

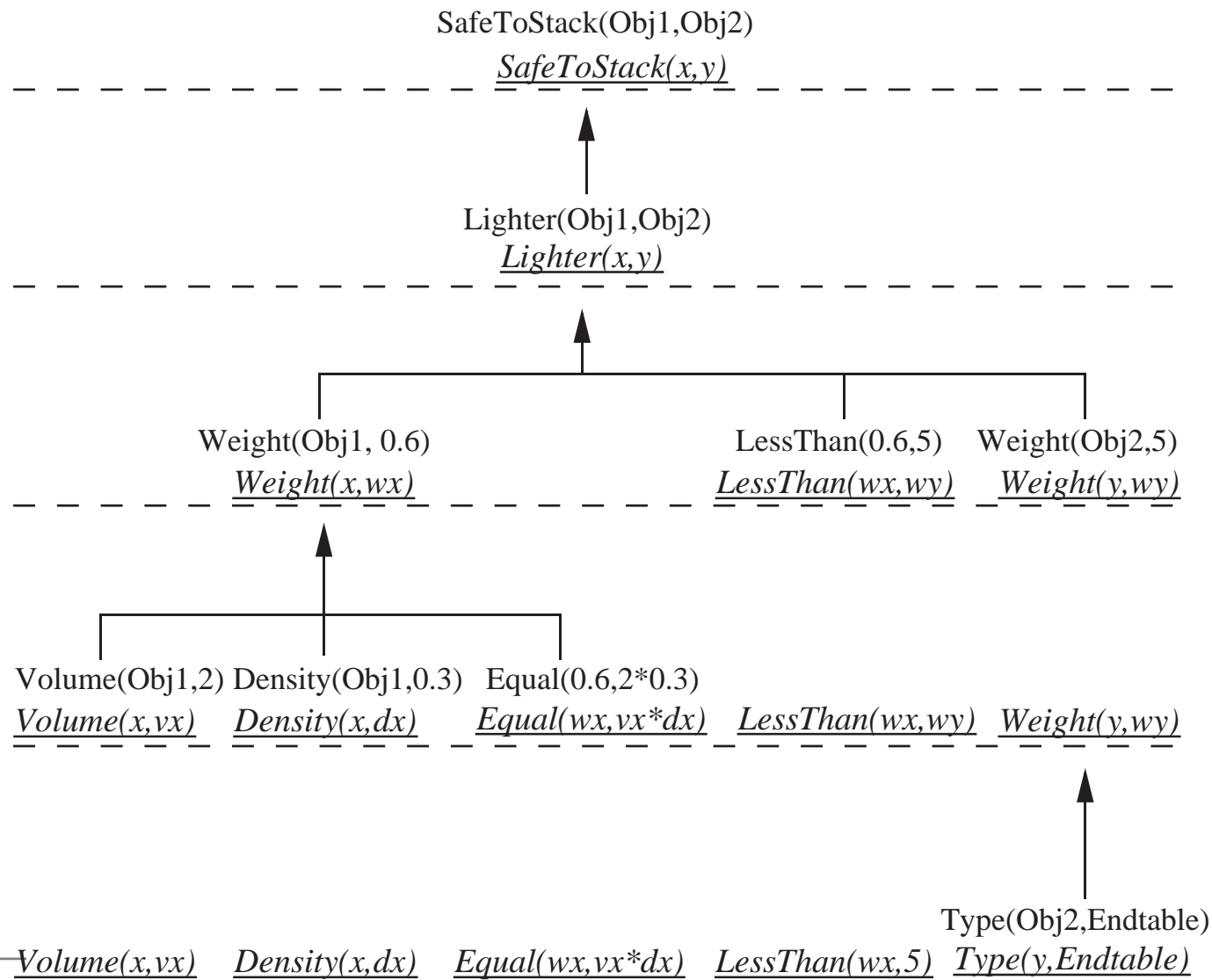


Training Example:



Computing Weakest Preimage

Computing Weakest Preimage of $SafeToStack(Obj1, Obj2)$



Regression

$\text{Regress}(\text{Frontier}, \text{Rule}, \text{Expression}, U_{I,R})$

Frontier: the set of expressions to be regressed through *Rule*

Rule: a horn clause.

Expression: the member of *Frontier* that is inferred by *Rule* in the explanation.

$U_{I,R}$: the substitution that unifies *Rule* to the training example in the explanation

Returns the list of expressions forming the weakest preimage of *Frontier* with respect to *Rule*

let Consequent \leftarrow *Rule* consequent

let Antecedents \leftarrow *Rule* antecedents

1. $U_{E,R} \leftarrow$ most general unifier of *Expression* with Consequent
such that there exists a substitution S for which

$$S(U_{E,R}(\text{Consequent})) = U_{I,R}(\text{Consequent})$$

2. Return $U_{E,R}(\{\text{Frontier} - \text{Consequent} + \text{Antecedents}\})$

Example:

```
Regress({Volume(x,vs), Density(x,dx), Equal(wx,vx*dx),  
        Less-Than(wx,wy), Weight(y,wy)},  
        Weight(z,5) :- Type(z,ENDTABLE),  
        Weight(y,wy),  
        {OBJ2/z})
```

Consequent \leftarrow Weight(z,5)

Antecedents \leftarrow Type(z,ENDTABLE)

$U_{E,R} \leftarrow \{y/z, 5/wy\}, (S = \{OBJ2/y\})$

Result $\leftarrow \{\text{Volume}(x,vs), \text{Density}(x,dx), \text{Equal}(wx,vx*dx),$
 $\text{Less-Than}(wx,5), \text{Type}(y,ENDTABLE)\}$

Regress

$\text{Regress}(\text{Frontier}, \text{Rule}, \text{Literal}, \theta_{hi})$

Frontier: literals to be regressed

Rule: A Horn clause

Literal: A literal in *Frontier* inferred by *Rule* in the explanation

θ_{hi} : The substitution that unifies the head of *Rule* to the corresponding literal in the explanation

Returns the set of literals forming the weakest preimage of *Frontier* with respect to *Rule*

- $head \leftarrow head \text{ of } Rule$
- $body \leftarrow body \text{ of } Rule$
- $\theta_{hl} \leftarrow$ the most general unifier of $head$ with *Literal* such that there exists a substitution θ_{li} for which

$$\theta_{li}(\theta_{hl}(head)) = \theta_{hi}(head)$$

- Return $\theta_{hl}(\text{Frontier} - head + body)$

Example

Regressing Literals

Example (the bottommost regression step):

Regress(*Frontier*, *Rule*, *Literal*, θ_{hi}) where

Frontier = {*Volume*(x, vs), *Density*(x, dx), *Eq*($wx, times(vx, dx)$),
LessThan(wx, wy), *Weight*(y, wy)}

Rule = *Weight*($z, 5$) \leftarrow *Type*($z, Endtable$)

Literal = *Weight*(y, wy)

$\theta_{hi} = \{z/Obj2\}$

- *head* \leftarrow *Weight*($z, 5$)
- *body* \leftarrow *Type*($z, Endtable$)
- $\theta_{hl} \leftarrow \{z/y, wy/5\}$, where $\theta_{li} = \{y/Obj2\}$
- **Return** {*Volume*(x, vs), *Density*(x, dx), *Eq*($wx, times(vx, dx)$),
LessThan($wx, 5$), *Type*($y, Endtable$)}

Inductive Bias

- Hypothesis follows *deductively* from $D \wedge B$
- Background theory B largely determines h
- Greedy search: preference for small sets of maximally general Horn clauses
- In contrast to inductive learning: bias is not a fixed, built-in property of the algorithm!

Inductive and Analytical Learning

	Inductive L.	Analytical L.
Goal	h fits data	h fits domain th.
Justification	statistical inference	deductive inference
Advantages	Little prior knowledge	learns from scarce data
pitfalls	scarce data, incorr. bias	imperfect domain th.

Related Work

- Combining induction and abduction (e.g. Dialogs)
- Generalization learning from analogy