Machine Learning

Analytical Learning

Artificial Intelligence & Computer Vision Lab
School of Computer Science and Engineering
Seoul National University

Overview

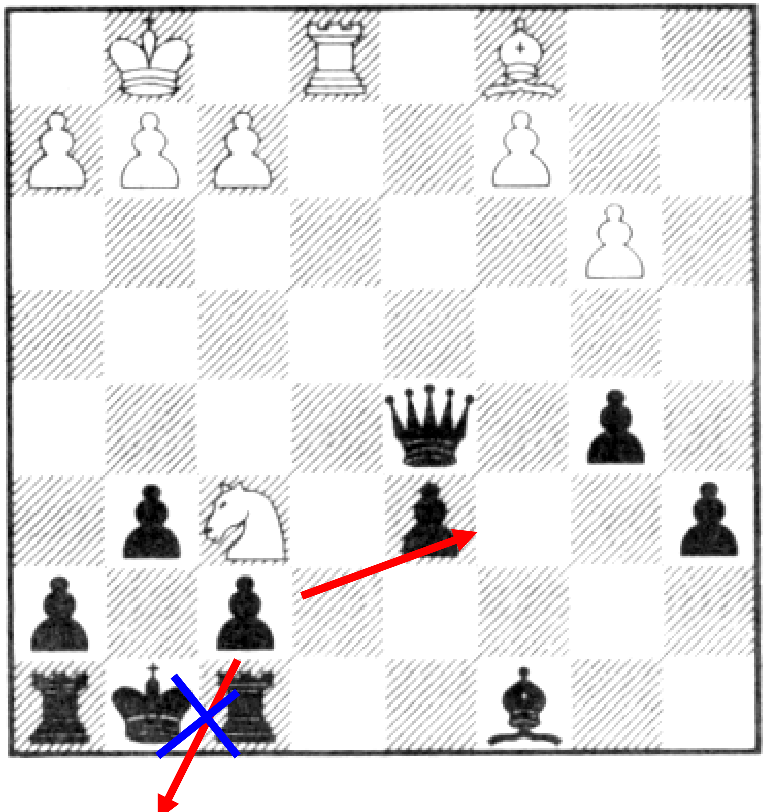
- Introduction
- Learning with Perfect Domain Theories : Prolog-EBG
- Remarks on Explanation-Based Learning

Introduction

- Inductive and analytical learning
  - Practical limit of inductive learning.
    - Require a certain number of training examples to achieve a given level of generalization accuracy.
    - They perform poorly when insufficient data is available.
  - Analytical learning
    - Use prior knowledge & deductive reasoning to argument the information provided by the training examples
    - It is not subject to fundamental bounds on learning accuracy imposed by the amount of training data available.
    - Explanation - based learning (EBL) :
      - Prior knowledge is used to analyze or explain how each observed training example satisfies the target concept. This explanation is then used to distinguish the relevant features of the training example from the irrelevant, so that examples are generalized based on this.

Introduction (cont.)

- Chess board example



Target Concept :

Chess board position in which black will lose its queen within two moves.

Inductive Learning :

Embarrassed, It's too complex !

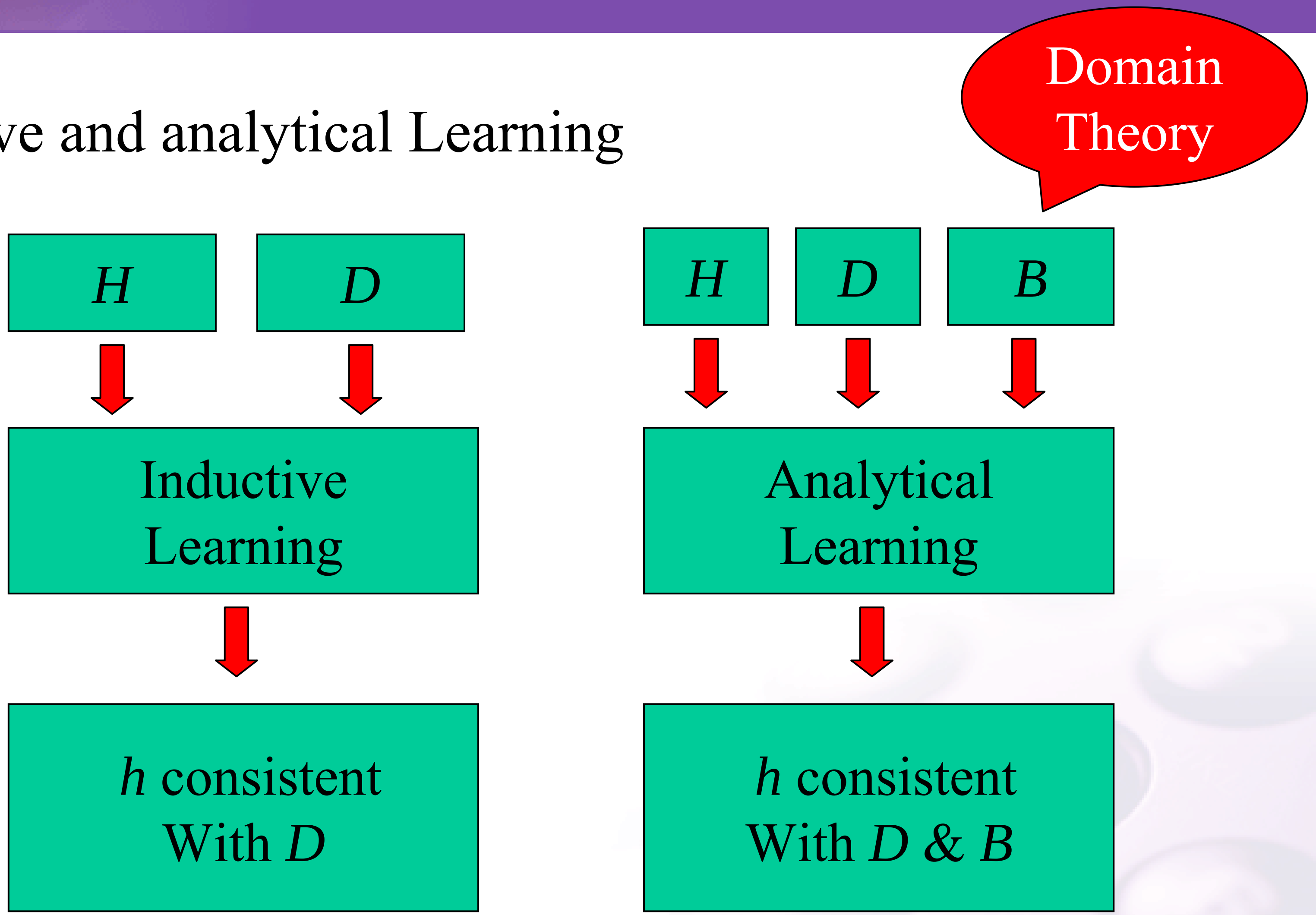
Analytical Learning :

We can explain the situation !

  - Board position : The black king and queen are simultaneously attacked.
  - Board position : Four white pawns are still in their original locations.

Introduction (cont.)

- Inductive and analytical Learning



Learning with Perfect Domain Theories : Prolog-EBG

- An illustrative trace
  - Explain the training example
  - Analyze the explanation
  - Refine the current hypothesis

Learning with Perfect Domain Theories : Prolog-EBG (cont.)

- Example – *SafeToStack*(*x*,*y*)
  - Given :
    - Instance space *X* : Each instance describes a pair of objects
    - Hypothesis space *H* : Each hypothesis is a set of Horn clause rules
    - Target concept : *SafeToStack*(*x*,*y*)
    - Training examples : typical positive example, *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), Color(Obj2,Blue), Material(Obj1,Cardboard), Material(Obj2,Wood), Volume(Obj1,2)
    - Domain theory *B* :

      SafeToStack(x,y) ← ¬ *Fragile(y)*

      SafeToStack(x,y) ← *Lighter(x,y)*

      Lighter(x,y) ← *Weight(x,wx)* ∧ *Weight(y,wy)* ∧ *LessThan(wx,wy)*

      Weight(x, w) ← *Volume(x,v)* ∧ *Density(x,d)* ∧ *Equal(w,times(v,d))*

      Weight(x,5) ← *Type(x,Endtable)*

      Fragile(x) ← *Material(x,Glass)*
    - …
  - Determine :
    - A hypothesis from *H* consistent with the training examples and domain theory.

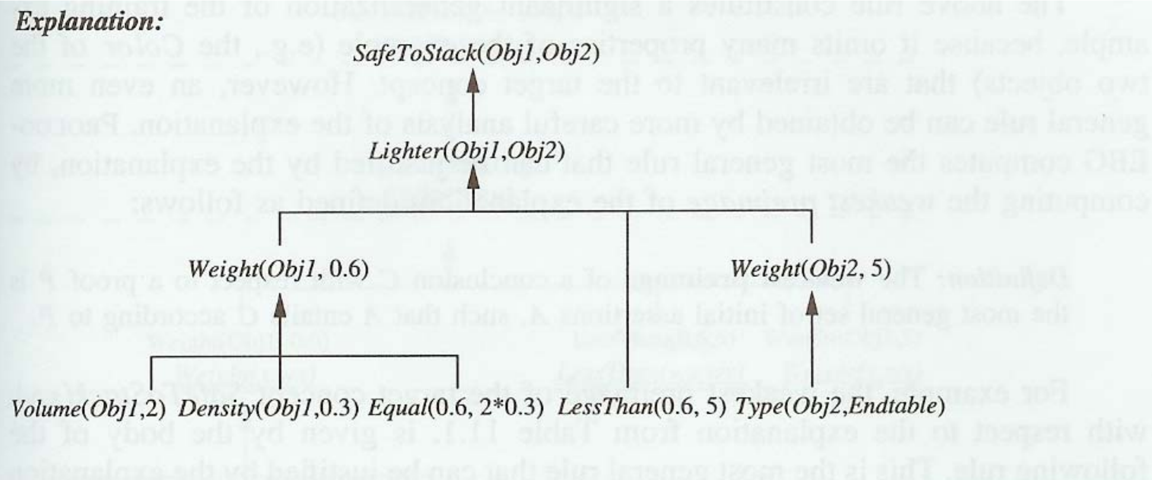
Learning with Perfect Domain Theories : Prolog-EBG (cont.)

- PROLOG-EBG(*TargetConcept*, *TrainingExamples*, *DomainTheroy*)
  - *LearnedRules* ← {}
  - *Pos* ← Positive examples from *TrainingExamples*
  - For each *PositiveExample* in *Pos* that is not covered by *LearnedRules*, do
    1. *Explain*:
       - *Explanation* ← An explanation (proof) in terms of the *DomainTheory* that *PositiveExample* satisfies the *TargetConcept*
    2. *Analyze*:
       - *SufficientConditions* ← The most general set of features of *PositiveExample* sufficient to satisfy the *TargetConcept* according to the *Explanation*
    3. *Refine*:
       - *LearnedRules* ← *LearnedRules* + *NewHornClause*, where *NewHornClause* is of the form

         TargetConcept ← *SufficientConditions*
  - Return *LearnedRules*

Learning with Perfect Domain Theories : Prolog-EBG (cont.)

- Explanation of a training example:



We can form a general rule that is justified by the domain theory…

$$SafeToStack(x,y) \leftarrow Volume(x,2) \wedge Density(x,0.3) \wedge Type(y,Endtable)$$

FIGURE 11.2
Explanation of a training example. The network at the bottom depicts graphically the training example *SafeToStack(Obj1, Obj2)* described in Table 11.1. The top portion of the figure depicts the explanation of how this example satisfies the target concept, *SafeToStack*. The shaded region of the training example indicates the example attributes used in the explanation. The other, irrelevant, example attributes will be dropped from the generalized hypothesis formed from this analysis.

Learning with Perfect Domain Theories : Prolog-EBG (cont.)

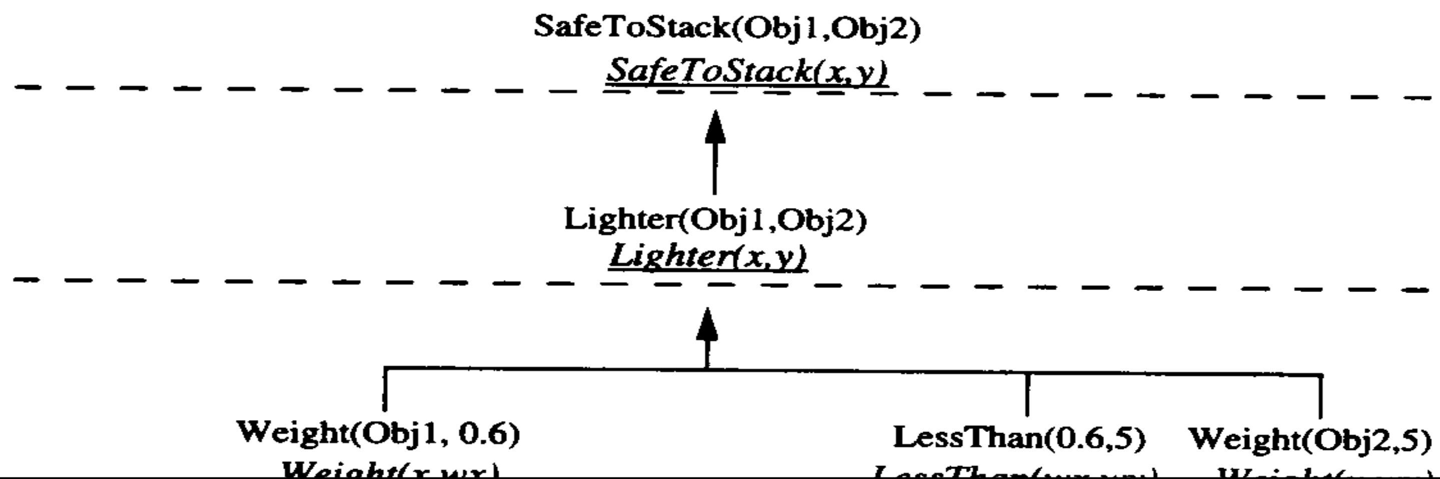
- Analyze the explanation:
  - Definition
    - The weakest preimage of a conclusion *C* with respect to a proof *P* is the most general set of initial assertion *A*, such that *A* entails *C* according to *P*.
  - Prolog – EBG computes the most general rule that can be justified by the explanation, by computing the weakest of the explanation.
  - Example: The *SafeToStack* problem.

$$SafeToStack(x,y) \leftarrow Volume(x,vx) \wedge Density(x,dx) \wedge Equal(wx,times(vx,dx)) \wedge LessThan(wx,5) \wedge Type(y,Endtable)$$

  - Prolog – EBG computes the weakest preimage of the target concept with respect to explanation, using regression procedure.

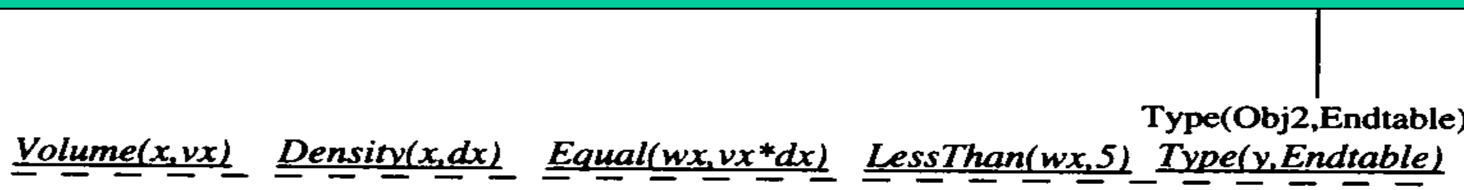
Learning with Perfect Domain Theories : Prolog-EBG (cont.)

Weakest preimage of *SafeToStack (obj1, obj2)*



Finally… we can get the most general rule

SafeToStack(x,y) ← *Volume(x,vx)* ∧ *Density(x,dx)* ∧ *Equal(wx,times(vx,dx))* ∧ *LessThan(wx,5)* ∧ *Type(y,Endtable)*



Learning with Perfect Domain Theories : Prolog-EBG (cont.)

- REGRESS (*Frontier, Rule, Literal,* θ_{hi})
  - *Frontier* : Set of literals to be regressed through *Rule*
  - *Rule* : A Horn clause
  - *Literal* : A *literal* in Frontier that is inferred by *Rule* in the *explanation*
  - Θ_{hi} : The substitution that unifies the *head* of *Rule* to the corresponding *literal* in the *explanation*
  - Return the set of *literals* forming the weakest preimage of *Frontier* with respect to *Rule*
    - *head* ← *head* of *Rule*
    - *body* ← *body* of *Rule*
    - Θ_{hl} ← 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 Θ_{hl} (*Frontier – head + body*)

Learning with Perfect Domain Theories : Prolog-EBG (cont.)

- Example:

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

$Frontier$ = { $Volume(x, vs), Density(d, dx), Equal\ (wx, times(vx, dx)),$
$LessThen(wx, wy), Weight(y, wy)$ }

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

$Literal = Weight(y,wy)$

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

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

Learning with Perfect Domain Theories : Prolog-EBG (cont.)

- Refine the current hypothesis:
  - At each stage, the sequential covering algorithm picks a new positive example not covered by the current Horn clauses, explains this new example, and formulates a new rule based on this.
  - Notice that only positive examples are covered in the algorithm. The learned set of rules predicts only positive examples and a new instance is classified as negative if it fails to predict that it is positive as in PROLOG

Remarks on Explanation-Based Learning

- Produces justified general hypotheses by using prior knowledge to analyze individual examples
- Explanation determines relevant attributes (features)
- Regressing allows deriving more general constraints (weakest preimage)
- Learned Horn clause corresponds to a sufficient condition to satisfy target concept
- Prolog-EBG implicitly assumes the complete and correct domain theory

Remarks on Explanation-Based Learning (cont.)

- Perspectives of EBL
  - EBL as theory-guided generalization of example
  - EBL as example-guided reformation of theories
  - EBL as “just” restating what the learner already “knows”

Remarks on Explanation-Based Learning (cont.)

- Discovering new features
  - Formulate new features not explicit in the training example
    - *SafeToStack* problem
    - The learned rule asserts that the product of Volume and Density of x is less than 5
    - The training examples contain no description of such a product
  - Similar to the features of hidden units of neural network
    - NN: Statistical process derives hidden unit features from many training examples
    - EBL: Analytical process derives features based on analysis of single examples using the domain theory

Remarks on Explanation-Based Learning (cont.)

- Deductive learning
  - Prolog-EBG outputs a hypothesis *h* that satisfies
    - $(\forall < x_i, f(x_i) > \in D)\ (h \wedge x_i) \vdash f(x_i)$
    - $D \wedge B \vdash h$
    - *B* : reduce the effective size of the hypothesis space
  - Inductive logic programming (Inverted deduction)
    - $(\forall < x_i, f(x_i) > \in D)\ (B' \wedge h \wedge x_i) \vdash f(x_i)$
    - ILP uses background knowledge *B'* to enlarge the set of hypotheses

Remarks on Explanation-Based Learning (cont.)

- Inductive bias in explanation-based learning
  - Approximate inductive bias of Prolog-EBG
    - Domain Theory *B*
    - Preference for small sets (sequential covering-algorithm) of maximally general (weakest preimage) Horn clauses
  - The property of Inductive Bias
    - Prolog-EBG : largely determined by input *B*
    - Other learning algorithm: determined by the syntax of hypothesis representation

⬇

  - Having domain-specific knowledge as input rather than restricting the syntax of hypothesis representation may be more natural to improve the generalization performance.
  - Thus, in considering the issue of how an autonomous agent may improve its learning capabilities over time, an agent with a learning algorithm whose generalization capabilities improve as it acquires more knowledge of its domain may be suggested.

Remarks on Explanation-Based Learning (cont.)

- Knowledge level learning
  - LEMMA-ENUMERATOR Algorithm
    - Ignores the training data and enumerates all proof tree
    - Calculates the weakest preimage and Horn clause like the Prolog-EBG
    - The output of Prolog-EBG $\subseteq$ The output of LEMMA-ENUMERATOR
  - The role of training examples
    - Focusing only on training examples encountered in practice
    - Develop a smaller, more relevant rules

Remarks on Explanation-Based Learning (cont.)

- Can Prolog-EBG learn a hypothesis that goes beyond the knowledge in domain theory?
  - Can't but it isn't a inherent limitation of deductive learning
  - Example of $B \nvdash h$ but $D \wedge B \vdash h$:

Domain theory (B) : $(\forall x)$ IF $((PlayTennis = Yes) \leftarrow (Humidity = x))$
THEN $((PlayTennis = Yes) \leftarrow (Humidity \leq x))$

$+$

Positive example (D) : $Humidity = .30$

$\downarrow$

Hypothesis (h) : $(PlayTennis = Yes) \leftarrow (Humidity \leq .30)$

  - Deductive closure of $B \subset$ Deductive closure of B+h