USING DOMAIN KNOWLEDGE IN MACHINE LEARNING:
Inductive vs. Analytical Learning

Carolina Ruiz
Motivation

Inductive Learning

induction
Examples -----------------> hypothesis
[+ Prior Knowledge] (generalization)

- Statistical reasoning is used to identify features that empirically distinguish $\oplus$ from $\odot$ examples.
  (e.g. decision trees, NNs, ILP, GAs)
- Fundamental bounds on accuracy depend on number of training examples.

Analytical Learning

deduction
Examples -----------------> hypothesis
+ Prior Knowledge (generalization)

- Logical reasoning is used to identify features that distinguish $\oplus$ from $\odot$ examples.
- Works well even when training examples are scarce.

Inductive + Analytical Learning

- Combines the best of both worlds
Learning

*Learning is to improve with experience at some task*

Learning from Examples: Inductive Learning

- **Given:** A set of training examples
  \[ \langle x_1, f(x_1) \rangle, \ldots, \langle x_n, f(x_n) \rangle \]
- **Find:** A hypothesis \( h \) defined over the whole space of instances that coincides with \( f \) over the training data, i.e. \( h(x_i) = f(x_i) \) for all \( 1 \leq i \leq n \).

Learning from Examples and Prior Knowledge

- **Given:** A set of training examples
  \[ \langle x_1, f(x_1) \rangle, \ldots, \langle x_n, f(x_n) \rangle \]
  and
  a set \( B \) of rules expressing prior knowledge
- **Find:** A hypothesis \( h \) defined over the whole space of instances s.t.
  \[ B \land h \land x_i \models f(x_i) \] for all \( 1 \leq i \leq n \).
Part 1: Motivation

Inductive Learning

induction

Examples ------------------> hypothesis
[+ Prior Knowledge] (generalization)

• Statistical reasoning is used to identify features that empirically distinguish $\oplus$ from $\odot$ examples. (e.g. decision trees, NNs, ILP, GAs)

• Fundamental bounds on accuracy depend on number of training examples.

• When the output hypothesis is a set of rules then this learning is called Inductive Logic Programming (ILP)

• ILP: uses prior knowledge to augment the input description of instances (which increases the complexity of the hypothesis space).
Inductive Logic Programming

INDUCTIVE LOGIC PROGRAMMING
   /
  /\ INDUCTIVE MACHINE LEARNING   COMPUTATIONAL LOGIC
   \

From Inductive Machine Learning: inherits its goal(s):

- to develop tools and techniques to induce hypotheses from observations (examples), or
- to synthesize new knowledge from experience.

From Computational Logic: inherits:

- ability to overcome limitations of classical inductive learners:
  1. use of limited knowledge representation formalism (e.g. propositional logic)
  2. inability to use substantial domain knowledge during learning process.
- theory, orientation, and techniques
- interest in properties of rules, convergence (soundness and completeness) of algorithms, computational complexity.
EXAMPLE (taken from Luc De Raedt and Nada Lavrac, 1993)

Given:

- a set of examples $D$
  \[ \oplus = \{ \text{flies(tweety)} \} \]
  \[ \ominus = \emptyset \]

- a background theory $B$
  \[ \text{bird(tweety)} \]
  \[ \text{bird(oliver)} \]

Find:

- a hypothesis $h$ that explains the examples $D$ w.r.t. $B$.
  \[ h = \{ \text{flies}(X) \leftarrow \text{bird}(X) \}. \]
EXAMPLE (taken from Luc De Raedt and Nada Lavrac, 1993)

Given:

- a set of examples $D$

$$
\oplus = \{ \text{sort}([1], [1]) \\
\quad \text{sort}([2, 1, 3], [1, 2, 3]) 
$$

$$
\otimes = \{ \text{sort}([2, 1], [1]) \\
\quad \text{sort}([3, 1, 2], [2, 1, 3]) 
$$

- a background theory $B$

$$
B = \{ \text{permutation}/2 \\
\quad \text{sorted}/1 
$$

Find:

- a hypothesis $h$ that satisfies the language bias and explains the examples $D$ w.r.t. $B$.

$$
h = \{ \text{sort}(X, Y) \leftarrow \text{permutation}(X, Y) \land \text{sorted}(Y) \}. $$
Sequential Covering Template

Given a set of examples $\oplus \cup \ominus :$

1. $\textit{learned-rules} := \{\}$

2. $\textbf{while}$ there are examples still to be explained $\textbf{do}$
   
   (a) $\textit{rule} := \text{Learn-one-rule}$

   (b) remove from examples those explained by $\textit{rule}$

   (c) $\textit{learned-rules} := \textit{learned-rules} \cup \textit{rule}$

3. Output $\textit{learned-rules}$
ILP implementations of Learn-one-rule

General to Specific Search (top down) e.g. FOIL

1. Start with the most general clause

   \[ e.g. \text{sort}(X, Y) \leftarrow \]

2. specialize the rule by adding the best possible literal to the body of the rule

   \[ e.g. \text{sort}(X, Y) \leftarrow \text{permutation}(X, Y) \]

   until the hypothesis entails no \( \varnothing \) examples ensuring that hypothesis entails at least a threshold number of \( \oplus \) examples.

Specific to General Search (bottom up) e.g. GOLEM

1. Start with the most specific clause that implies a given example

2. generalize

3. until the hypothesis cannot be further generalized without implying negative examples.
Searching for the best literal – Foil [Quinlan, 1990]

**Candidate Literals:** All literals that can be constructed using predicates and terms that appear in $B$ and/or in the examples.

**Evaluation Function:** Foil’s information-based notion of gain

**Note:** This search can be computationally explosive

**Important:** to prune the search space!

Matthew Berube’s MQP (advised by C. Ruiz and S. Alvarez) prunes the search space by using type information making FOIL applicable to some large domains.
Summarizing Inductive Learning

- Uses statistical inference to compute a hypothesis that fit the data

- Uses background knowledge to augment the input description of instances (which may increase the complexity of the hypothesis space).

+ Requires little background knowledge

- Requires large amounts of training data
Part II: Motivation

Analytical Learning

deduction
Examples \rightarrow \text{hypothesis} + \text{Prior Knowledge} \quad \text{(generalization)}

- Logical reasoning is used to identify features that distinguish $\oplus$ from $\ominus$ examples.
- Works well even when training examples are scarce.
- Analytical Learning Method: Explanation–Based Learning (EBL)
- EBL: uses prior knowledge to reduce the complexity of the hypothesis space and to improve the accuracy of the output hypothesis.
- Prior Knowledge is assumed to be correct and complete
Analytical Learning

The Analytical Learning Task:

- **Given:** A set $D$ of training examples $\langle x_i, f(x_i) \rangle$ and Background Knowledge $B$ (represented as a set of rules / Horn clauses)

- **Find:** a hypothesis $h$ such that:
  1. $D \land B \vdash h$
  2. for all $1 \leq i \leq n$, $h \land x_i \vdash f(x_i)$

Part (1) above reduces the size of the hypothesis space and makes the output hypothesis more accurate.

**Assumptions:** The background knowledge is

- **Correct:** each of its assertions is a truthful statement about the world
- **Complete:** every instance that satisfies the target concept can be proven by the domain knowledge to satisfy it.

Are these reasonable assumptions? Why bother learning if the background knowledge is already complete?
Analytical Learning – Motivation

• Chess example (taken from Mitchell’s book)

• Many features are present. Prior knowledge helps us tell apart relevant features from irrelevant ones.

• For the chess domain, it is easy to write down rules that completely describe all legal moves, but it is very hard to write down rules that completely describe an optimal move strategy.
**EBL: An Analytical Learning Method**

**EBL: Explanation-Based Learning**

- Prior knowledge is used to construct an explanation/proof of each example (which is expressed as a Horn rule)

- this explanation is used to distinguish between the relevant features of the training example and the irrelevant ones

- the explanation (rule) is generalized to the extent possible and added to the current hypothesis (set of rules)

- representative of several EBL algorithms
- guaranteed to output a hypothesis that is correct and covers the positive training examples
- sequential covering algorithm:

1. hypothesis := {}
2. For each positive training example not explained by hypothesis do
   (a) **Explain**, using the background knowledge, why the example satisfies the target concept
      (uses Prolog’s backtracking)
   (b) **Analyze** the explanation to determine the most general conditions under which the explanation holds
      (uses regression)
   (c) **Refine** the hypothesis by adding this explanation to it.
3. Output hypothesis
Prolog-EBG: Example (taken from Mitchell’s book)
EBL: Different Perspectives

• Using the background knowledge to generalize the training examples
  rational/logical generalization

• Using the training examples to reformulate the background knowledge
  reformulating B into a more operational form
  rules in the hypothesis classify an instance in a single inference step → knowledge compilation

• Using B and D to restate what the learner already knows
  difference between what one knows in principle and what one can efficiently do in practice
Summarizing Analytical Learning

- Uses logical deduction to compute a hypothesis that fit the data and the background knowledge

- Uses background knowledge to augment the information provided by the training data (which decreases the size of the hypothesis space).

  - Requires correct and complete background knowledge

  + Does well on scarce data
PART III: Motivation

Inductive + Analytical Learning

Induction + deduction
Examples \[\rightarrow\] hypothesis
[+ Prior Knowledge] (generalization)

The Learning Task:

- **Given:** A set $D$ of training examples $\langle x_i, f(x_i) \rangle$ and Background Knowledge $B$ (represented as a set of rules / Horn clauses)
- **Find:** a hypothesis $h$ that best fits $D$ and $B$.

Assumptions:

**Data:** May contain errors, may be scarce

**Prior Knowledge:** If available, it may be incorrect or incomplete

**Goal:** Do better than inductive or analytical learning alone
Inductive + Analytical Search for $h$

- **Use $B$ to derive an initial hypothesis from which to begin the search**
  
  e.g. KBANN [Shavlik & Towell 1989]

- **Use $B$ to refine the hypothesis search space**
  
  e.g. EBNN [Mitchell & Thrun 1993], Matt Berube’s MQP

- **Use $B$ to alter the available search steps**
  
  e.g. FOCL [Pazzani et al. 1991]