Lecture 9

Analytical Learning Discussion (4 of 4): Integrating Inductive and Analytical Learning

Monday, February 7, 2000

William H. Hsu
Department of Computing and Information Sciences, KSU
http://www.cis.ksu.edu/~bhsu

Readings:
Mitchell, Chapter 2
Russell and Norvig, Chapter 21

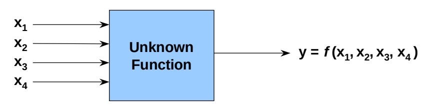
Lecture Outline

- References: Chapters 2-3, Mitchell
- Suggested Exercises: 2.2, 2.3, 2.4, 2.6
- Review: Learning from Examples
  - (Supervised) concept learning framework
  - Basic inductive learning algorithms
- General-to-Specific Ordering over Hypotheses
  - Version space: partially-ordered set (poset) formalism
  - Candidate elimination algorithm
  - Inductive learning
- Decision Trees
  - Quick tutorial / review: Lectures 4-5, CIS 798, Fall 1999
  - See: http://ringil.cis.ksu.edu/Courses/Fall-1999/CIS798/Lectures
- Relation to Analytical Learning
- Next Class: Introduction to Artificial Neural Networks

(Supervised) Concept Learning

- Given: Training Examples <*x*, *f(x)*> of Some Unknown Function *f*
- Find: A Good Approximation to *f*
- Examples (besides Concept Learning)
  - Disease diagnosis
    - *x* = properties of patient (medical history, symptoms, lab tests)
    - *f* = disease (or recommended therapy)
  - Risk assessment
    - *x* = properties of consumer, policyholder (demographics, accident history)
    - *f* = risk level (expected cost)
  - Automatic steering
    - *x* = bitmap picture of road surface in front of vehicle
    - *f* = degrees to turn the steering wheel
  - Part-of-speech tagging
  - Fraud/intrusion detection
  - Web log analysis
  - Multisensor integration and prediction

A Learning Problem



Example	x_1	x_2	x_3	x_4	y
0	0	1	1	0	0
1	0	0	0	0	0
2	0	0	1	1	1
3	1	0	0	1	1
4	0	1	1	0	0
5	1	1	0	0	0
6	0	1	0	1	0

- x_i: t_i, y: t, *f*: $(t_1 \times t_2 \times t_3 \times t_4) \to t$
- Our learning function: Vector $(t_1 \times t_2 \times t_3 \times t_4 \times t) \to ((t_1 \times t_2 \times t_3 \times t_4) \to t)$

Hypothesis Space: Unrestricted Case

- $| A \to B | = | B |^{|A|}$
- $|H^4 \to H| = | \{0,1\} \times \{0,1\} \times \{0,1\} \times \{0,1\} \to \{0,1\} | = 2^{2^4} = 65536$ function values
- Complete Ignorance: Is Learning Possible?
  - Need to see every possible input/output pair
  - After 7 examples, still have $2^9 = 512$ possibilities (out of 65536) for *f*

Example	x_1	x_2	x_3	x_4	y
0	0	0	0	0	?
1	0	0	0	1	?
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	?
8	1	0	0	0	?
9	1	0	0	1	1
10	1	0	1	0	?
11	1	0	1	1	?
12	1	1	0	0	0
13	1	1	0	1	?
14	1	1	1	0	?
15	1	1	1	1	?

Training Examples for Concept *EnjoySport*

- Specification for Examples
  - Similar to a data type definition
  - 6 attributes: Sky, Temp, Humidity, Wind, Water, Forecast
  - Nominal-valued (symbolic) attributes - enumerative data type
- Binary (Boolean-Valued or H-Valued) Concept
- Supervised Learning Problem: *Describe the General Concept*

Example	Sky	Air Temp	Humidity	Wind	Water	Forecast	Enjoy Sport
0	Sunny	Warm	Normal	Strong	Warm	Same	**Yes**
1	Sunny	Warm	High	Strong	Warm	Same	**Yes**
2	Rainy	Cold	High	Strong	Warm	Change	**No**
3	Sunny	Warm	High	Strong	Cool	Change	**Yes**

Representing Hypotheses

- Many Possible Representations
- Hypothesis *h*: Conjunction of Constraints on Attributes
- Constraint Values
  - Specific value (e.g., *Water = Warm*)
  - Don't care (e.g., *"Water = ?"*)
  - No value allowed (e.g., "Water = Ø")
- Example Hypothesis for *EnjoySport*
  - Sky AirTemp Humidity Wind Water Forecast
    <Sunny ? ? Strong ? Same>
  - Is this consistent with the training examples?
  - What are some hypotheses that are consistent with the examples?

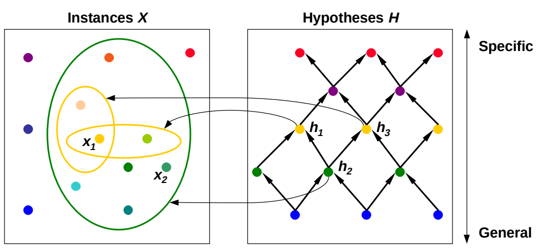
Prototypical Concept Learning Tasks

- Given
  - Instances X: possible days, each described by attributes *Sky, AirTemp, Humidity, Wind, Water, Forecast*
  - Target function *c* ≡ EnjoySport: $X \to$ H ≡ {{Rainy, Sunny} × {Warm, Cold} × {Normal, High} × {None, Mild, Strong} × {Cool, Warm} × {Same, Change}} → {0, 1}
  - Hypotheses *H*: conjunctions of literals (e.g., *<?, Cold, High, ?, ?, ?>*)
  - Training examples *D*: positive and negative examples of the target function

$$\langle x_1, c(x_1)\rangle, \ldots, \langle x_m, c(x_m)\rangle$$

- Determine
  - Hypothesis $h \in H$ such that $h(x) = c(x)$ for all $x \in D$
  - Such *h* are consistent with the training data
- <u>What Is A Concept Learning Algorithm?</u>
  - L: Vector ($X \times$ H ≡ boolean) → ($X \to$ H)
  - Type of L means: given vector of examples (data set), return <u>hypothesis</u> *h*
  - *h*: $X \to$ H

Instances, Hypotheses, and the Partial Ordering *Less-Specific-Than*



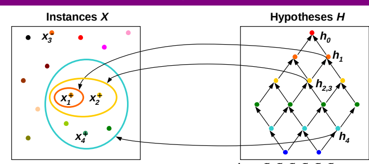
x_1 = *<Sunny, Warm, High, Strong, Cool, Same>*
x_2 = *<Sunny, Warm, High, Light, Warm, Same>*

h_1 = *<Sunny, ?, ?, Strong, ?, ?>*
h_2 = *<Sunny, ?, ?, ?, ?, ?>*
h_3 = *<Sunny, ?, ?, ?, Cool, ?>*

$\leq_P$ ≡ *Less-Specific-Than* ≡ *More-General-Than*

$h_2 \leq_P h_1$
$h_2 \leq_P h_3$

Hypothesis Space Search by *Find-S*



x_1 = *<Sunny, Warm, Normal, Strong, Warm, Same>, +*
x_2 = *<Sunny, Warm, High, Strong, Warm, Same>, +*
x_3 = *<Rainy, Cold, High, Strong, Warm, Change>, -*
x_4 = *<Sunny, Warm, High, Strong, Cool, Change>, +*

h_1 = *<Ø, Ø, Ø, Ø, Ø, Ø>*
h_2 = *<Sunny, Warm, Normal, Strong, Warm, Same>*
h_3 = *<Sunny, Warm, ?, Strong, Warm, Same>*
h_4 = *<Sunny, Warm, ?, Strong, Warm, Same>*
h_5 = *<Sunny, Warm, ?, Strong, ?, ?>*

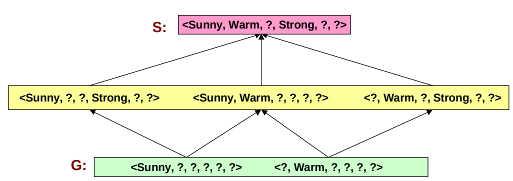
- Shortcomings of *Find-S*
  - Can't tell whether it has learned concept
  - Can't tell when training data inconsistent
  - Picks a maximally specific *h* (why?)
  - Depending on *H*, there might be several!

Version Spaces

- Definition: Consistent Hypotheses
  - A hypothesis *h* is <u>consistent</u> with a set of training examples *D* of target concept *c* if and only if $h(x) = c(x)$ for each training example *<x, c(x)>* in *D*.
  - *Consistent* (*h*, *D*) ≡ $\forall$ *<x, c(x)>* $\in$ *D* . $h(x) = c(x)$
- Definition: Version Space
  - The <u>version space</u> $VS_{H,D}$, with respect to hypothesis space *H* and training examples *D*, is the subset of hypotheses from *H* consistent with all training examples in *D*.
  - $VS_{H,D} \equiv \{ h \in H \mid Consistent(h, D) \}$

The *List-Then-Eliminate* Algorithm

1. Initialization: *VersionSpace* ← a list containing every hypothesis in *H*
2. For each training example *<x, c(x)>*
   Remove from *VersionSpace* any hypothesis *h* for which $h(x) \neq c(x)$
3. Output the list of hypotheses in *VersionSpace*



Example Version Space

Representing Version Spaces

- **Hypothesis Space**
  - A finite *meet semilattice* (partial ordering Less-Specific-Than; ⊥ ≡ all *?*)
  - Every pair of hypotheses has a *greatest lower bound* (GLB)
  - $VS_{H,D}$ ≡ the consistent poset (partially-ordered subset of *H*)
- **Definition: General Boundary**
  - General boundary *G* of version space $VS_{H,D}$: set of most general members
  - Most general ≡ *minimal* elements of $VS_{H,D}$ ≡ "set of necessary conditions"
- **Definition: Specific Boundary**
  - Specific boundary S of version space $VS_{H,D}$: set of most specific members
  - Most specific ≡ *maximal* elements of $VS_{H,D}$ ≡ "set of sufficient conditions"
- **Version Space**
  - Every member of the version space lies between *S* and *G*
  - $VS_{H,D} \equiv \{ h \in H \mid \exists s \in S . \exists g \in G . g \leq_P h \leq_P s \}$ where $\leq_P$ ≡ Less-Specific-Than

Candidate Elimination Algorithm [1]

1. Initialization

   G ← (singleton) set containing most general hypothesis in *H*, denoted {<?, … , ?>}

   S ← set of most specific hypotheses in *H*, denoted {<Ø, … , Ø>}

2. For each training example *d*

   If *d* is a positive example (*Update-S*)

   Remove from *G* any hypotheses inconsistent with *d*

   For each hypothesis *s* in *S* that is not consistent with *d*

   Remove *s* from *S*

   Add to *S* all minimal generalizations *h* of *s* such that

   1. *h* is consistent with *d*
   2. Some member of *G* is more general than *h*

   (These are the greatest lower bounds, or *meets*, $s \vee d$, in $VS_{H,D}$)

   Remove from *S* any hypothesis that is more general than another hypothesis in *S* (remove any dominated elements)

Candidate Elimination Algorithm [2]

(continued)

If *d* is a negative example (*Update-G*)

Remove from *S* any hypotheses inconsistent with *d*

For each hypothesis *g* in *G* that is not consistent with *d*

Remove *g* from *G*

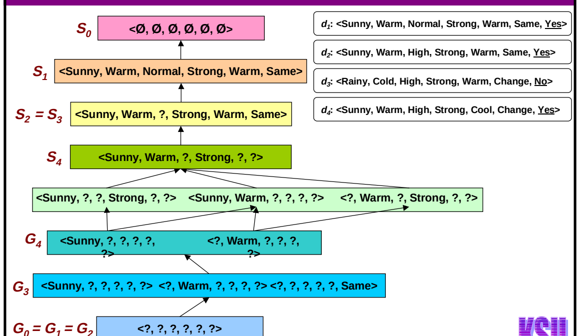
Add to *G* all minimal specializations *h* of *g* such that

1. *h* is consistent with *d*
2. Some member of *S* is more specific than *h*

(These are the least upper bounds, or *joins*, $g \wedge d$, in $VS_{H,D}$)

Remove from *G* any hypothesis that is less general than another hypothesis in *G* (remove any dominating elements)

Example Trace



Terminology

- **Supervised Learning**
  - Concept – function: observations to categories; so far, boolean-valued (+/-)
  - Target (function) – true function *f*
  - Hypothesis – proposed function *h* believed to be similar to *f*
  - Hypothesis space – space of all hypotheses that can be generated by the learning system
  - Example – tuples of the form <*x, f(x)*>
  - Instance space (*aka* example space) – space of all possible examples
  - Classifier – discrete-valued function whose range is a set of class labels
- **Inductive Learning**
  - Inductive generalization – process of generating hypotheses $h \in H$ that describe cases not yet observed
  - The inductive learning hypothesis – basis for inductive generalization
- **Analytical Learning**
  - Domain theory *T* – set of assertions to *explain* examples
  - Analytical generalization - process of generating *h* consistent with *D* and *T*
  - Explanation – proof in terms of *T* that *x* satisfies *c*(*x*)

Summary Points

- **Concept Learning as Search through *H***
  - Hypothesis space *H* as a state space
  - Learning: finding the correct hypothesis
- **Inductive Leaps Possible Only if Learner Is Biased**
  - Futility of learning without bias
  - Strength of inductive bias: proportional to restrictions on hypotheses
- **Modeling Inductive Learners**
  - Equivalent inductive learning, deductive inference (theorem proving) problems
  - Hypothesis language: syntactic restrictions (*aka* representation bias)
- **Views of Learning and Strategies**
  - Removing uncertainty ("data compression")
  - Role of knowledge
- **Integrated Inductive and Analytical Learning**
  - Using inductive learning to acquire domain theories for analytical learning
  - Roles of integrated learning in KDD
- **Next Time: Introduction to ANNs**