Lecture 2

Inductive Bias and PAC Learning

Tuesday, August 31, 1999

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

Readings:
Sections 2.7-2.8, Sections 7.1-7.3, Mitchell
Sections 2.4.1-2.4.3, Shavlik and Dietterich

Lecture Outline

- Read 2.7-2.8, 7.1-7.3, Mitchell; 2.4.1-2.4.3 S&D
- Homework 1: Due Thursday, September 16, 1999 (before 5 PM CST)
- Paper Commentary 1: Due This Thursday
  - L. G. Valiant, "A Theory of the Learnable" (*Communications of the ACM*, 1984)
  - See guidelines in course notes
- The Need for Inductive Bias
  - Representations (hypothesis languages): a worst-case scenario
  - Change of representation
- Computational Learning Theory
  - Setting 1: learner poses queries to teacher
  - Setting 2: teacher chooses examples
  - Setting 3: randomly generated instances, labeled by teacher
- Probably Approximately Correct (PAC) Learning
  - Motivation
  - Introduction to PAC framework

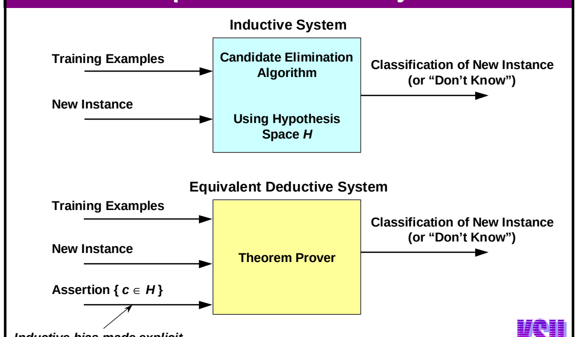
An Unbiased Learner

- Example of A Biased *H*
  - *Conjunctive* concepts with don't cares
  - What concepts can *H* not express? (Hint: what are its syntactic limitations?)
- Idea
  - Choose *H'* that expresses every teachable concept
  - i.e., *H'* is the power set of *X*
  - Recall: $| \mathcal{A} \rightarrow \mathcal{B} | = | \mathcal{B} |^{|\mathcal{A}|}$ ($\mathcal{A} = X$; $\mathcal{B}$ = {labels}; $H' = \mathcal{A} \rightarrow \mathcal{B}$)
  - {{Rainy, Sunny} × {Warm, Cold} × {Normal, High} × {None, Mild, Strong} × {Cool, Warm} × {Same, Change}} → {0, 1}
- An Exhaustive Hypothesis Language
  - Consider: *H'* = disjunctions ($\vee$), conjunctions ($\wedge$), negations ($\neg$) over previous *H*
  - $| H' | = 2^{(2 \cdot 2 \cdot 2 \cdot 3 \cdot 2 \cdot 2)} = 2^{96}$; $| H | = 1 + (3 \cdot 3 \cdot 3 \cdot 4 \cdot 3 \cdot 3) = 973$
- What Are S, G For The Hypothesis Language *H'*?
  - S ← *disjunction of all positive examples*
  - G ← *conjunction of all negated negative examples*

Inductive Bias

- Components of An Inductive Bias Definition
  - Concept learning algorithm *L*
  - Instances *X*, target concept *c*
  - Training examples $D_c = \{<x, c(x)>\}$
  - $L(x_i, D_c)$ = classification assigned to instance x_i by *L* after training on D_c
- Definition
  - The inductive bias of *L* is any minimal set of assertions *B* such that, for any target concept c and corresponding training examples D_c,
    $$\forall x_i \in X \,.\, [(B \wedge D_c \wedge x_i) \vdash L(x_i, D_c)]$$
    where A ⊢ B means A *logically entails* B
  - Informal idea: preference for (i.e., restriction to) certain hypotheses by structural (syntactic) means
- Rationale
  - Prior assumptions regarding target concept
  - Basis for inductive generalization

Inductive Systems and Equivalent Deductive Systems

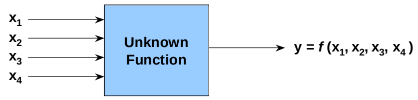


Three Learners with Different Biases

- Rote Learner
  - Weakest bias: anything seen before, i.e., no bias
  - Store examples
  - Classify *x if and only if* it matches previously observed example
- Version Space Candidate Elimination Algorithm
  - Stronger bias: concepts belonging to conjunctive *H*
  - Store extremal generalizations and specializations
  - Classify *x if and only if* it "falls within" *S* and *G* boundaries (all members agree)
- Find-S
  - Even stronger bias: most specific hypothesis
  - Prior assumption: any instance *not observed to be positive* is negative
  - Classify *x* based on *S* set

Hypothesis Space: A Syntactic Restriction

- **Recall: 4-Variable Concept 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

- **Bias: Simple Conjunctive Rules**
  - Only 16 simple conjunctive rules of the form $y = x_i \wedge x_j \wedge x_k$
  - $y = \emptyset, x_1, \ldots, x_4, x_1 \wedge x_2, \ldots, x_3 \wedge x_4, x_1 \wedge x_2 \wedge x_3, \ldots, x_2 \wedge x_3 \wedge x_4, x_1 \wedge x_2 \wedge x_3 \wedge x_4$
  - Example above: no simple rule explains the data (counterexamples?)
  - Similarly for simple clauses (conjunction and disjunction allowed)

Hypothesis Space: m-of-n Rules

- ***m-of-n* Rules**
  - 32 possible rules of the form:
    "y = 1 iff at least m of the following n variables are 1"

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

	Counterexample					Counterexample			
Variables	1-of	2-of	3-of	4-of	Variables	1-of	2-of	3-of	4-of
$\{x_1\}$	2	–	–	–	$\{x_2, x_4\}$	1	2	–	–
$\{x_2\}$	1	–	–	–	$\{x_3, x_4\}$	3	3	–	–
$\{x_3\}$	0	–	–	–	$\{x_1, x_2, x_3\}$	0	2	2	–
$\{x_4\}$	6	–	–	–	$\{x_1, x_2, x_4\}$	1	2	2	–
$\{x_1, x_2\}$	1	2	–	–	$\{x_1, x_3, x_4\}$	0	•	2	–
$\{x_1, x_3\}$	0	2	–	–	$\{x_2, x_3, x_4\}$	0	4	2	–
$\{x_1, x_4\}$	5	2	–	–	$\{x_1, x_2, x_3, x_4\}$	0	4	2	2
$\{x_2, x_3\}$	1	2	–	–					

- **Found A Consistent Hypothesis!**

Views of Learning

- **Removal of (Remaining) Uncertainty**
  - Suppose unknown function was *known* to be *m-of-n* Boolean function
  - Could use training data to infer the function
- **Learning and Hypothesis Languages**
  - Possible approach to *guess a good, small hypothesis language*:
    - Start with a very small language
    - Enlarge until it contains a hypothesis that fits the data
  - Inductive bias
    - Preference for certain languages
    - Analogous to data compression (removal of redundancy)
    - Later: coding the "model" versus coding the "uncertainty" (error)
- **We Could Be Wrong!**
  - Prior knowledge could be wrong (e.g., $y = x_4 \wedge$ one-of (x_1, x_3) also consistent)
  - If guessed language was wrong, errors will occur on new cases

Two Strategies for Machine Learning

- **Develop Ways to Express Prior Knowledge**
  - Role of prior knowledge: guides search for hypotheses / hypothesis languages
  - Expression languages for prior knowledge
    - Rule grammars; stochastic models; etc.
    - Restrictions on computational models; other (formal) specification methods
- **Develop Flexible Hypothesis Spaces**
  - Structured collections of hypotheses
    - Agglomeration: nested collections (hierarchies)
    - Partitioning: decision trees, lists, rules
    - Neural networks; cases, etc.
  - Hypothesis spaces of adaptive size
- **Either Case: Develop Algorithms for Finding A Hypothesis That Fits Well**
  - Ideally, will generalize well
- **Later: Bias *Optimization* (Meta-Learning, Wrappers)**

Computational Learning Theory

- **What General Laws Constrain Inductive Learning?**
- **What Learning Problems Can Be Solved?**
- **When Can We Trust The Output of A Learning Algorithm?**
- **We Seek Theory To Relate:**
  - Probability of successful learning
  - Number of training examples
  - Complexity of hypothesis space
  - Accuracy to which target concept is approximated
  - Manner in which training examples are presented

Prototypical Concept Learning Task

- **Given**
  - Instances X: possible days, each described by attributes *Sky, AirTemp, Humidity, Wind, Water, Forecast*
  - Target function $c \equiv$ EnjoySport: $X \to \mathcal{H}$
  - Hypotheses H: conjunctions of literals, e.g.,

    <?, Cold, High, ?, ?, ?>
  - Training examples D: positive and negative examples of the target function

    $<x_1, c(x_1)>, <x_2, c(x_2)>, \ldots, <x_m, c(x_m)>$
- **Determine**
  - A hypothesis h in H such that $h(x) = c(x)$ for all x in D?
  - A hypothesis h in H such that $h(x) = c(x)$ for all x in X?

Sample Complexity

- How Many Training Examples Sufficient To Learn Target Concept?
- Scenario 1: Active Learning
  - Learner proposes instances, as *queries* to teacher
  - Query (learner): instance *x*
  - Answer (teacher): *c*(*x*)
- Scenario 2: Passive Learning from Teacher-Selected Examples
  - Teacher (who knows *c*) provides training examples
  - Sequence of examples (teacher): {<x_i, $c(x_i)$>}
  - Teacher *may or may not* be helpful, optimal
- Scenario 3: Passive Learning from Teacher-Annotated Examples
  - Random process (e.g., nature) proposes instances
  - Instance *x* generated randomly, teacher provides *c*(*x*)

Sample Complexity: Scenario 1

- Learner Proposes Instance *x*
- Teacher Provides *c*(*x*)
  - *Comprehensibility*: assume c is in learner's hypothesis space *H*
  - A form of inductive bias (sometimes nontrivial!)
- Optimal Query Strategy: Play 20 Questions
  - Pick instance *x* such that half of hypotheses in *VS* classify x positive, half classify *x* negative
  - When this is possible, need $\lceil \log_2 |H| \rceil$ queries to learn *c*
  - When not possible, need even more

Sample Complexity: Scenario 2

- Teacher Provides Training Examples
  - Teacher: agent who knows *c*
  - Assume *c* is in learner's hypothesis space *H* (as in Scenario 1)
- Optimal Teaching Strategy: Depends upon *H* Used by Learner
  - Consider case: *H* = conjunctions of up to *n* boolean literals and their negations
  - e.g., (*AirTemp* = *Warm*) $\wedge$ (*Wind* = *Strong*), where *AirTemp*, *Wind*, … each have 2 possible values
  - Complexity
    - If *n* possible boolean attributes in *H*, *n* + 1 examples suffice
    - Why?

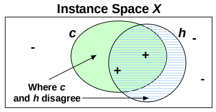
Sample Complexity: Scenario 3

- Given
  - Set of instances *X*
  - Set of hypotheses *H*
  - Set of possible target concepts *C*
  - Training instances generated by a *fixed*, unknown probability distribution $\mathcal{D}$ over *X*
- Learner Observes Sequence *D*
  - *D*: training examples of form <*x*, *c*(*x*)> for target concept *c* $\in$ *C*
  - Instances *x* are drawn from distribution $\mathcal{D}$
  - Teacher provides target value *c*(*x*) for each
- Learner Must Output Hypothesis *h* Estimating *c*
  - *h* evaluated on performance on subsequent instances
  - Instances still drawn according to $\mathcal{D}$
- Note: Probabilistic Instances, Noise-Free Classifications

True Error of A Hypothesis

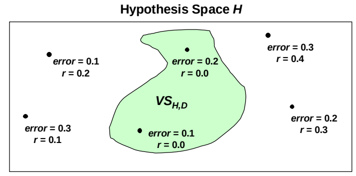
- Definition
  - The <u>true error</u> (denoted $error_{\mathcal{D}}(h)$) of hypothesis *h* with respect to target concept *c* and distribution $\mathcal{D}$ is the probability that *h* will misclassify an instance drawn at random according to $\mathcal{D}$.
  - $error_{\mathcal{D}}(h) \equiv \Pr_{x \in \mathcal{D}}[c(x) \neq h(x)]$
- Two Notions of Error
  - <u>Training error</u> of hypothesis *h* with respect to target concept *c*: how often $h(x) \neq c(x)$ over training instances
  - <u>True error</u> of hypothesis *h* with respect to target concept *c*: how often $h(x) \neq c(x)$ over future random instances
- Our Concern
  - Can we bound true error of *h* (given training error of *h*)?
  - First consider when training error of *h* is zero (i.e, $h \in VS_{H,D}$)

Instance Space *X*



Exhausting The Version Space

Hypothesis Space *H*



(*r* = training error, *error* = true error)

- Definition
  - The version space $VS_{H,D}$ is said to be ε-<u>exhausted</u> with respect to *c* and $\mathcal{D}$, if every hypothesis *h* in $VS_{H,D}$ has error less than ε with respect to *c* and $\mathcal{D}$.
  - $\forall\, h \in VS_{H,D}$. $error_{\mathcal{D}}(h) < \varepsilon$

Number of Examples Required to Exhaust The Version Space

- How Many Examples Will ε–Exhaust The Version Space?
- Theorem [Haussler, 1988]
  - If the hypothesis space H is finite, and D is a sequence of $m \geq 1$ independent random examples of some target concept c, then for any $0 \leq \varepsilon \leq 1$, the probability that the version space with respect to H and D is not ε-exhausted (with respect to c) is less than or equal to

$$|H| \, e^{-\varepsilon m}$$

- Important Result!
  - *Bounds the probability* that any consistent learner will output a hypothesis h with $error(h) \geq \varepsilon$
  - Want this probability to be below a specified threshold δ

$$|H| \, e^{-\varepsilon m} \leq \delta$$

  - To achieve, solve inequality for m: let

$$m \geq 1/\varepsilon \, (\ln |H| + \ln (1/\delta))$$

  - Need to see *at least* this many examples

Learning Conjunctions of Boolean Literals

- How Many Examples Are Sufficient?
  - Specification - ensure that with probability *at least* $(1 - \delta)$:

$$\text{Every } h \text{ in } VS_{H,D} \text{ satisfies } error_{\mathcal{D}}(h) < \varepsilon$$

  - "The probability of an ε-bad hypothesis ($error_{\mathcal{D}}(h) \geq \varepsilon$) is no more than δ"
  - Use our theorem:

$$m \geq 1/\varepsilon \, (\ln |H| + \ln (1/\delta))$$

  - H: conjunctions of constraints on up to n boolean attributes (n boolean literals)
  - $|H| = 3^n$, $m \geq 1/\varepsilon \, (\ln 3^n + \ln (1/\delta)) = 1/\varepsilon \, (n \ln 3 + \ln (1/\delta))$
- How About *EnjoySport*?
  - H as given in *EnjoySport* (conjunctive concepts with don't cares)
    - $|H| = 973$
    - $m \geq 1/\varepsilon \, (\ln |H| + \ln (1/\delta))$
  - Example goal: probability $1 - \delta$ = 95% of hypotheses with $error_{\mathcal{D}}(h) < 0.1$
  - $m \geq 1/0.1 \, (\ln 973 + \ln (1/0.05)) \approx 98.8$

PAC Learning

- Terms Considered
  - Class C of possible concepts
  - Set of instances X
  - Length n (in attributes) of each instance
  - Learner L
  - Hypothesis space H
  - Error parameter (error bound) ε
  - Confidence parameter (excess error probability bound) δ
  - *size*(c) = the encoding length of c, assuming some representation
- Definition
  - C is PAC-learnable by L using H if for all $c \in C$, distributions $\mathcal{D}$ over X, ε such that $0 < \varepsilon < 1/2$, and δ such that $0 < \delta < 1/2$, learner L will, with probability at least $(1 - \delta)$, output a hypothesis $h \in H$ such that $error_{\mathcal{D}}(h) \leq \varepsilon$
  - C is efficiently PAC-learnable if L runs in time polynomial in $1/\varepsilon$, $1/\delta$, n, *size*(c)

Agnostic Learning

- Assumption of "Knowable" Concept
  - So far, assumed $c \in H$
  - Agnostic learning environment: don't assume $c \in H$
- What Do We Want Then?
  - The "closest hypothesis we can get"
  - Hypothesis h that makes the fewest errors on training data
- How Hard Is This?
  - Sample complexity:

$$m \geq 1/2\varepsilon^2 \, (\ln |H| + \ln (1/\delta))$$

  - Derived from Hoeffding bounds:

$$P\,[error_{\mathcal{D}}(h) > error_D(h) + \varepsilon] \leq e^{-2m\varepsilon^2}$$

Terminology

- Inductive Bias
  - Strength of inductive bias: how *few* hypotheses?
  - Specific biases: based on specific languages
- Hypothesis Language
  - "Searchable subset" of the space of possible descriptors
  - *m-of-n*, conjunctive, disjunctive, clauses
  - Ability to represent a concept
- PAC Learning
  - Probably Approximately Correct
  - Computational Learning Theory (COLT)
  - True error versus training error
  - Notation: distribution $\mathcal{D}$, $error_{\mathcal{D}}(h)$, ε-bad with probability δ
  - ε-exhaustion: every hypothesis in $VS_{H,D}$ has $error_{\mathcal{D}}(h) < \varepsilon$
  - PAC-learnability: for $c \in C$, X, n, L, H, ε, δ

Summary Points

- 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 with Equivalent Deductive Systems
  - Representing inductive learning as theorem proving
  - Equivalent learning and inference problems
- Syntactic Restrictions
  - Example: *m-of-n* concept
- Views of Learning and Strategies
  - Removing uncertainty ("data compression")
  - Role of knowledge
- Introduction to Computational Learning Theory (COLT)
  - Things COLT attempts to measure
  - Probably-Approximately-Correct (PAC) learning framework
- Next Lecture: Occam's Razor, VC Dimension, and Error Bounds