



Lecture 01 of 42

Introduction to Machine Learning and the Version Space Formalism

Wednesday, 24 January 2008

William H. Hsu

Department of Computing and Information Sciences, KSU

KSOL course pages: <http://snipurl.com/1y5gc> / <http://snipurl.com/1y5ih>
Course web site: <http://www.kddresearch.org/Courses/Spring-2008/CIS732>
Instructor home page: <http://www.cis.ksu.edu/~bhsu>

Reading for Next Class:

Syllabus and Course Intro

Handout: Chapters 1-2, Mitchell (at K-State Union Copy Center)



Computing & Information Sciences
Kansas State University



Review: Course Administration

- **Course Pages (KSOL):** <http://snipurl.com/1y5gc> / <http://snipurl.com/1y5ih>
- **Class Web Page:** www.kddresearch.org/Courses/Spring-2008/CIS732
- **Instructional E-Mail Addresses (for Topics in AI, substitute 830 for 732)**
 - * CIS732TA-L@listserv.ksu.edu (always use this to reach instructor and TA)
 - * CIS732-L@listserv.ksu.edu (this goes to everyone)
- **Instructor: William Hsu, Nichols 213**
 - * Office phone: +1 785 532 7905; home phone: +1 785 539 7180
 - * IM: AIM/MSN/YIM [hsuw@rizenabsith](#), ICQ [28651394/191317559](#), Google [banazir](#)
 - * Office hours: after class Mon/Wed/Fri; other times by appointment
- **Graduate Teaching Assistant: Jing Xia**
 - * Office location: Nichols 213a
 - * Office hours: to be announced on class web board
- **Grading Policy**
 - * Midterm: 15% (in-class, open-book); final (take-home): 20%
 - * Machine problems, problem sets (6 of 8): 30%; term project: 20%
 - * Paper reviews (10 of 12): 10%; class participation: 5% (HW, Q&A)



Computing & Information Sciences
Kansas State University



Learning Functions

- **Notation and Definitions**
 - * **Instance:** $\vec{x} = (x_1, x_2, \dots, x_n)$, sometimes \vec{x}_j , $1 \leq j \leq m$ with x_{ji} , $1 \leq i \leq n$
 - * **Instance space** X such that $x \in X$
 - * **Data set:** $D = \{x_1, x_2, \dots, x_m\}$ where $x_j = (x_{j1}, x_{j2}, \dots, x_{jn})$
- **Clustering**
 - * Mapping from old $x = (x_1, x_2, \dots, x_n)$ to new $x' = (x'_1, x'_2, \dots, x'_k)$, $k \ll n$
 - * Attributes x'_i of new instance not necessarily named
 - * Idea: project instance space X into lower dimension X'
 - * Goal: keep groups of similar X together in X'
- **Regression**
 - * Idea: given independent variable x , dependent variables $y = f(x)$, fit f
 - * Goal: given new (previously unseen) x , approximate $f(x)$
- **Classification**
 - * Similar to regression, except that f is boolean- or nominal-valued
 - * "Curve fitting" figurative – approximator may be logical formula
- **Skills**
 - * Evaluation functions
 - * Policies



Prototypical Concept Learning Tasks

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

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

- **Determine**
 - * Hypothesis $h \in H$ such that $h(x) = c(x)$ for all $x \in D$
 - * Such h are consistent with the training data
- **Training Examples**
 - * Assumption: no missing X values
 - * Noise in values of c (contradictory labels)?



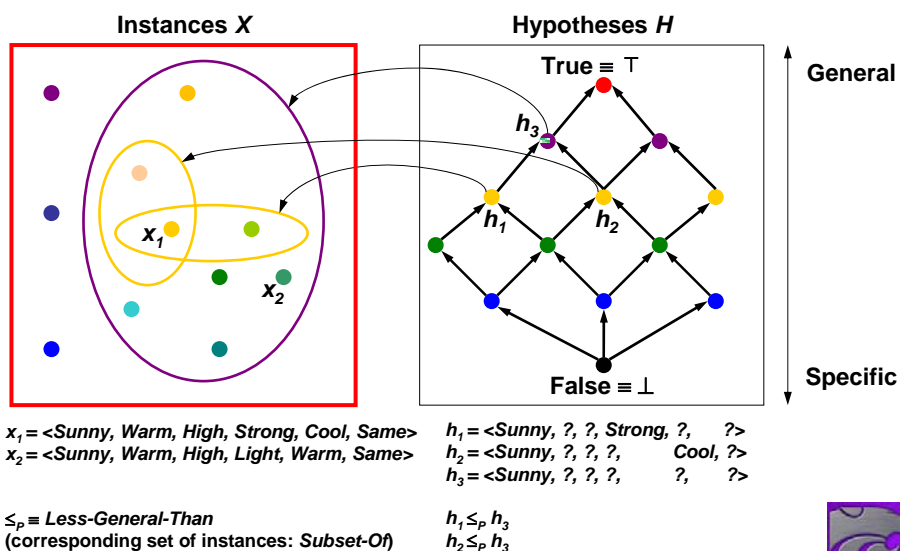


The Inductive Learning Hypothesis

- **Fundamental Assumption of Inductive Learning**
- **Informal Statement**
 - * Any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over other unobserved examples
 - * Definitions deferred: *sufficiently large, approximate well, unobserved*
- **Later: Formal Statements, Justification, Analysis**
 - * Different operational definitions – see: Chapters 5 – 7, Mitchell (1997)
 - * Statistical: sufficient statistics
 - * Probabilistic: distributions for training and validation/test data
 - * Computational: sample complexity, confidence and error bounds
- **Next: How to Find This Hypothesis?**



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





Find-S Algorithm

1. Initialize h to the most specific hypothesis in H

H : the hypothesis space

(partially ordered set under relation *Less-Specific-Than*)

2. For each positive training instance x

For each attribute constraint a_i in h

IF constraint a_i in h is satisfied by x

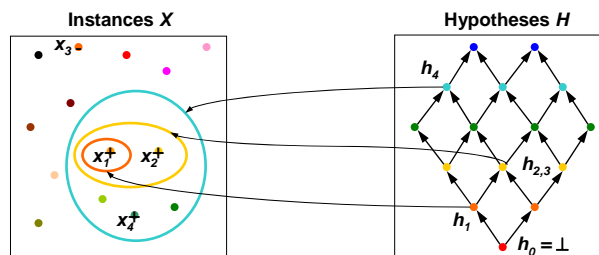
THEN do nothing

ELSE replace a_i in h by next more general constraint satisfied by x

3. Output hypothesis h



Hypothesis Space Search by Find-S



$x_1 = \langle \text{Sunny, Warm, Normal, Strong, Warm, Same} \rangle, +$
 $x_2 = \langle \text{Sunny, Warm, High, Strong, Warm, Same} \rangle, +$
 $x_3 = \langle \text{Rainy, Cold, High, Strong, Warm, Change} \rangle, -$
 $x_4 = \langle \text{Sunny, Warm, High, Strong, Cool, Change} \rangle, +$

$h_0 = \langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle$
 $h_1 = \langle \text{Sunny, Warm, Normal, Strong, Warm, Same} \rangle$
 $h_2 = \langle \text{Sunny, Warm, ?, Strong, Warm, Same} \rangle$
 $h_3 = \langle \text{Sunny, Warm, ?, Strong, Warm, Same} \rangle$
 $h_4 = \langle \text{Sunny, Warm, ?, Strong, ?, ?} \rangle$

● 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 consistent with a set of training examples D of target concept c if and only if $h(x) = c(x)$ for each training example $\langle x, c(x) \rangle$ in D .
- * $Consistent(h, D) \equiv \forall \langle x, c(x) \rangle \in D. h(x) = c(x)$

- **Given**

- * Hypothesis space H
- * Data set D : set of training examples

- **Definition**

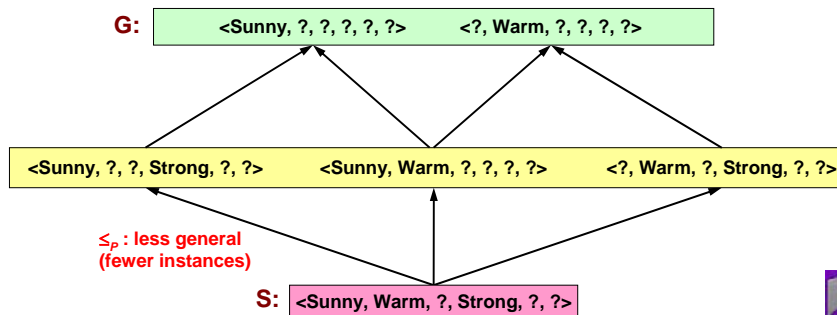
- * Version space $VS_{H,D}$ with respect to H, D
- * Subset of hypotheses from H consistent with all training examples in D
- * $VS_{H,D} \equiv \{ h \in H \mid Consistent(h, D) \}$



List-Then-Eliminate Algorithm

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

Example Version Space





Hypothesis Spaces As Lattices

- **Meet Semilattice**
 - * Every pair of hypotheses h_i and h_j has *greatest lower bound (GLB)* $h_i \vee h_j$
 - * Is H meet semilattice?
 - * $\perp \equiv \text{some } \emptyset$
- **Join Semilattice**
 - * Every pair of hypotheses h_i and h_j has *least upper bound (LUB)* $h_i \wedge h_j$
 - Is H join semilattice?
 - * $\top \equiv \text{all ?}$
- **(Full) Lattice**
 - * Every pair of hypotheses has GLB $h_i \vee h_j$ and LUB $h_i \wedge h_j$
 - * Both meet semilattice and join semilattice
 - * Partial ordering Less-General-Than



Representing Version Spaces As Lattices

- **Definition: General (Upper) Boundary**
 - * General boundary G of version space $VS_{H,D}$: set of most general members
 - * Most general \equiv *maximal* elements of $VS_{H,D}$ \equiv "set of necessary conditions"
- **Definition: Specific (Lower) Boundary**
 - * Specific boundary S of version space $VS_{H,D}$: set of least general members
 - * Most specific \equiv *minimal* elements of $VS_{H,D}$ \equiv "set of sufficient conditions"
- **Version Space**
 - * $VS_{H,D} \equiv$ consistent poset (partially-ordered subset of H)
 - * Every member of version space lies between S and G
 - * $VS_{H,D} \equiv \{ h \in H \mid \exists s \in S, g \in G. s \leq_p h \leq_p g \}$, $\leq_p \equiv$ Less-General-Than

"Version space is defined as set of hypotheses sandwiched between specific s and general g (given data)"





Candidate Elimination Algorithm [1]

1. Initialization

$G_0 \leftarrow \top \equiv$ most general hypothesis in H , denoted $\{<?, \dots, ?>\}$

$S_0 \leftarrow \perp \equiv$ least general hypotheses in H , denoted $\{<\emptyset, \dots, \emptyset>\}$

2. For each training example d

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

Remove from G any hypotheses inconsistent with d

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

Remove s from S // "move S upwards"

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 least upper bounds, or *joins*, $s \wedge d$, in $VS_{H,D}$)

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



Candidate Elimination Algorithm [2]

(continued)

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

Remove from S any hypotheses inconsistent with d

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

Remove g from G // "move G downwards"

Add to G all minimal specializations h of g such that

1. h is consistent with d
2. Some member of S is less general than h

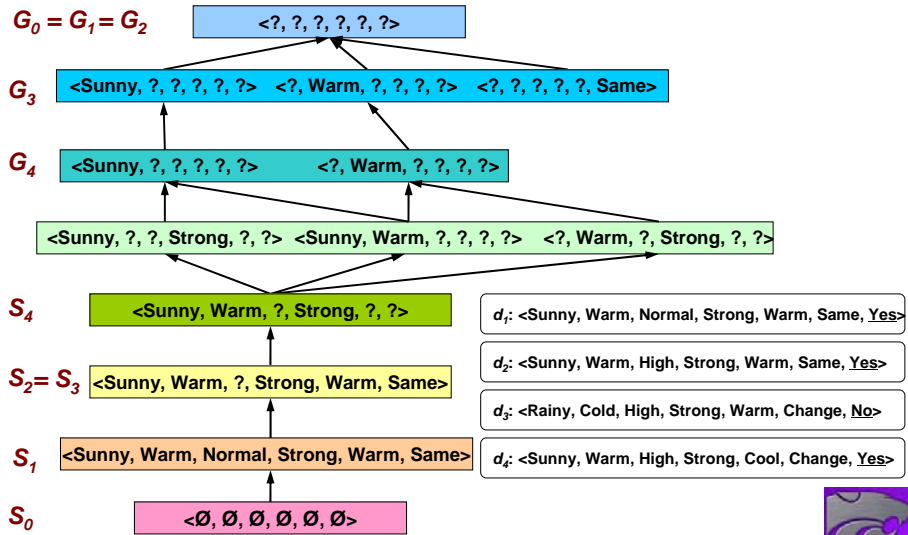
(These are greatest lower bounds, or *meets*, $g \vee d$, in $VS_{H,D}$)

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

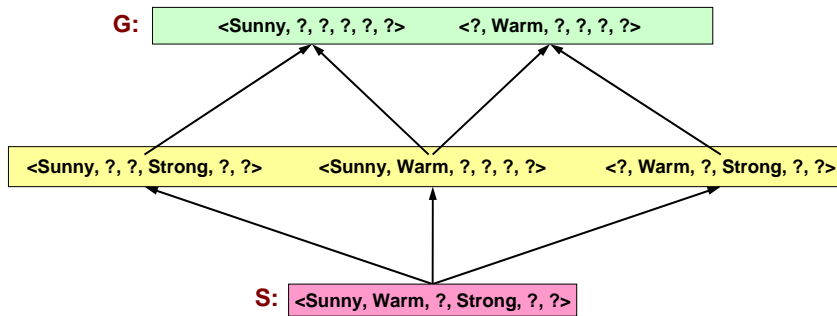




Example Trace



What Next Training Example?



- **Active Learning: What Query Should The Learner Make Next?**
- **How Should These Be Classified?**
 - * <Sunny, Warm, Normal, Strong, Cool, Change>
 - * <Rainy, Cold, Normal, Light, Warm, Same>
 - * <Sunny, Warm, Normal, Light, Warm, Same>



What Justifies This Inductive Leap?

- **Example: Inductive Generalization**
 - * Positive example: <Sunny, Warm, Normal, Strong, Cool, Change, Yes>
 - * Positive example: <Sunny, Warm, Normal, Light, Warm, Same, Yes>
 - * Induced S: <Sunny, Warm, Normal, ?, ?, ?>
- **Why Believe We Can Classify The Unseen?**
 - * e.g., <Sunny, Warm, Normal, Strong, Warm, Same>
 - * When is there enough information (in a new case) to make a prediction?



An Unbiased Learner

- **Inductive Bias**
 - Any preference for one hypothesis over another, *besides* consistency
 - Example: $H \equiv$ conjunctive concepts with don't cares
 - What concepts can H not express? (Hint: what are its syntactic limitations?)
- **Idea**
 - Choose unbiased H' : expresses every teachable concept (i.e., power set of X)
 - Recall: $|A \rightarrow B| = |B|^{|A|}$ ($A = X$; $B = \{\text{labels}\}$; $H' = A \rightarrow B$)
 - $\{\{\text{Rainy, Sunny, Cloudy}\} \times \{\text{Warm, Cold}\} \times \{\text{Normal, High}\} \times \{\text{None-Mild, Strong}\} \times \{\text{Cool, Warm}\} \times \{\text{Same, Change}\}\} \rightarrow \{0, 1\}$
- **An Exhaustive Hypothesis Language**
 - Consider: H' = disjunctions (\vee), conjunctions (\wedge), negations (\neg) over 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' ?**
 - $S \leftarrow$ disjunction of all positive examples
 - $G \leftarrow$ conjunction of all negated negative examples





Summary Points

- **Concept Learning as Search through H**
 - * Hypothesis space H as a state space
 - * Learning: finding the correct hypothesis
- **General-to-Specific Ordering over H**
 - * Partially-ordered set: Less-Specific-Than (More-General-Than) relation
 - * Upper and lower bounds in H
- **Version Space Candidate Elimination Algorithm**
 - * S and G boundaries characterize learner's uncertainty
 - * Version space can be used to make predictions over unseen cases
- **Learner Can Generate Useful Queries**
- **Next Lecture: When and Why Are Inductive Leaps Possible?**



Terminology

- **Hypotheses**
 - * Classification function or classifier – nominal-valued function f
 - * Regression function or regressor – integer or real-valued function f
 - * Hypothesis – proposed function h believed to be similar to c (or f)
 - * Instance / unlabeled example – tuples of the form $x = (x_1, x_2, \dots, x_m)$
 - * Example / labeled instance – tuples of the form $\langle x, f(x) \rangle$
- **The Version Space Algorithm**
 - * Consistent hypothesis - one that correctly predicts observed examples
 - * Version space - space of all currently consistent (or *satisfiable*) hypotheses
 - * Meet semilattice (GLB \vee), join semilattice (LUB \wedge), lattice (GLB and LUB)
 - * Algorithms: *Find-S*, *List-Then-Eliminate*, candidate elimination
- **Inductive Learning**
 - * Inductive generalization - process of generating hypotheses that describe cases not yet observed
 - * Inductive learning hypothesis – when generalization is feasible
 - * Inductive bias – any preference for one h over another *other* than consistency with data D

