Computational Learning and Discovery

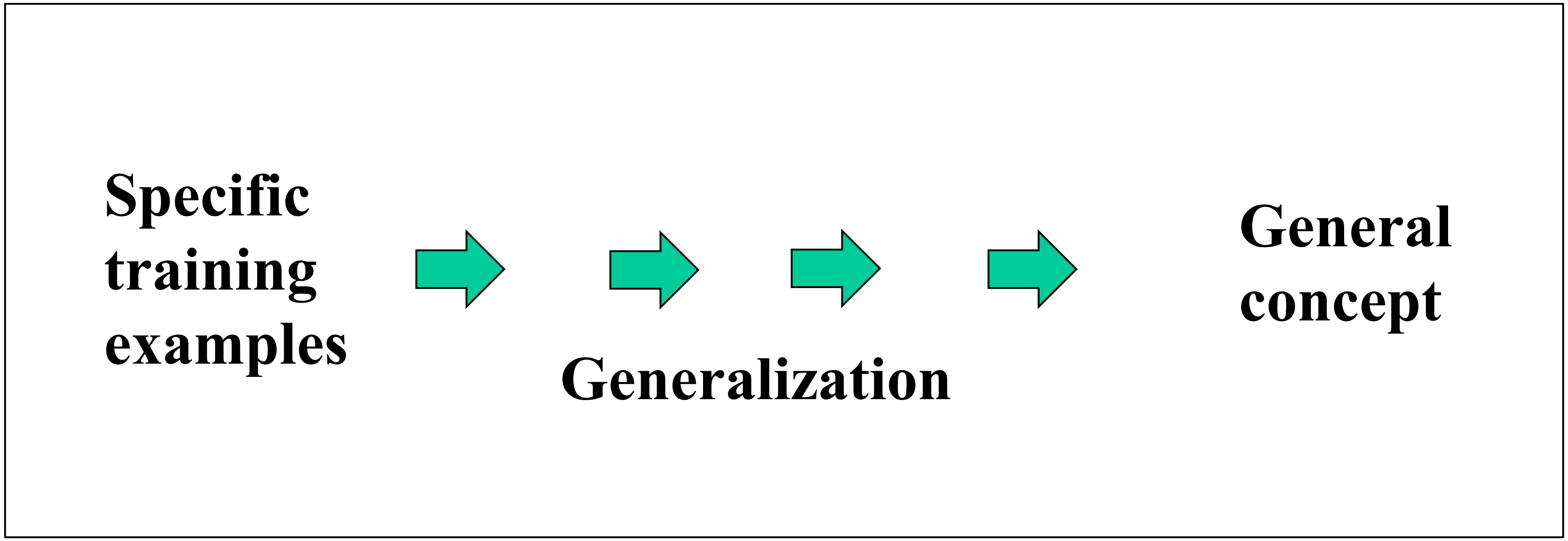
CSI 873 / MATH 689

Instructor: I. Griva

Wednesday 7:20 – 10:00 pm

Concept learning and general to specific ordering

Learning



Concept learning-

acquiring the definition of a general category given a sample of positive and negative training examples

Concept learning-

acquiring the definition of a general category given a sample of positive and negative training examples

can be formulated as a problem of searching through a predefined space of potential hypotheses for the hypothesis that best fits the training examples.

Concept learning-

acquiring the definition of a general category given a sample of positive and negative training examples

can be formulated as a problem of searching through a predefined space of potential hypotheses for the hypothesis that best fits the training examples.

Inferring a boolean-valued function from training examples of its input and output

Concept learning - example

Learning the target concept “days on which my friend Aldo enjoys his favorite water sport”

Concept learning - example

Learning the target concept “days on which my friend Aldo enjoys his favorite water sport”

Formalizing…

Training examples are represented by a set of attributes: *Sky, AirTemp, Humidity, Wind, Water, Forecast, and EnjoySport.*

Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

Example of the hypothesis (*?,Cold,High,?,?,?*)

The goal is find a hypothesis the closest to target concept

Notations

? indicates that any value is acceptable

$\emptyset$ indicates that no value is acceptable

The most general hypothesis: (?,?,?,?,?,?)

The most specific hypothesis: ($\emptyset$, $\emptyset$, $\emptyset$, $\emptyset$, $\emptyset$, $\emptyset$)

More general notations

X is a set of instances over which the concept is defined

c is a target concept, $c: X \rightarrow \{0, 1\}$

$(x, c(x))$ is a training example (could be negative or positive)

H is the set of all possible hypotheses the learner considers to identify the target concept

The goal is to find a hypothesis h in H such that $h(x) = c(x)$ for all x in X

Inductive learning hypothesis:

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 data

How large is the Hypothesis Space?

Sky: ***Sunny, Cloudy*** **and** ***Rainy***
AirTemp: ***Warm*** **and** ***Cold***
Humidity: ***Normal*** **and** ***High***
Wind: ***Strong*** **and** ***Weak***
Water: ***Warm*** **and** ***Cool***
Forecast: ***Same*** **and** ***Change***

General-to-Specific Ordering of Hypotheses

$$h_1 = (Sunny, ?, ?, Strong, ?, ?)$$

$$h_2 = (Sunny, ?, ?, ?, ?, ?)$$

h_2 is more general than h_1

Definition

Let h_j and h_k be boolean - valued functions defined over X.

Then h_j is more_general_than_or_equal_to h_k ($h_j \geq_g h_k$)

if and only if

$$\forall x \in X \left[(h_k(x) = 1) \rightarrow (h_j(x) = 1) \right]$$

General-to-Specific Ordering of Hypotheses

Definition

Let h_j and h_k be boolean - valued functions defined over X.

Then h_j is more_general_than_or_equal_to h_k ($h_j \geq_g h_k$)

if and only if

$$\forall x \in X \left[(h_k(x) = 1) \rightarrow (h_j(x) = 1) \right]$$

Let h_j and h_k be boolean - valued functions defined over X.

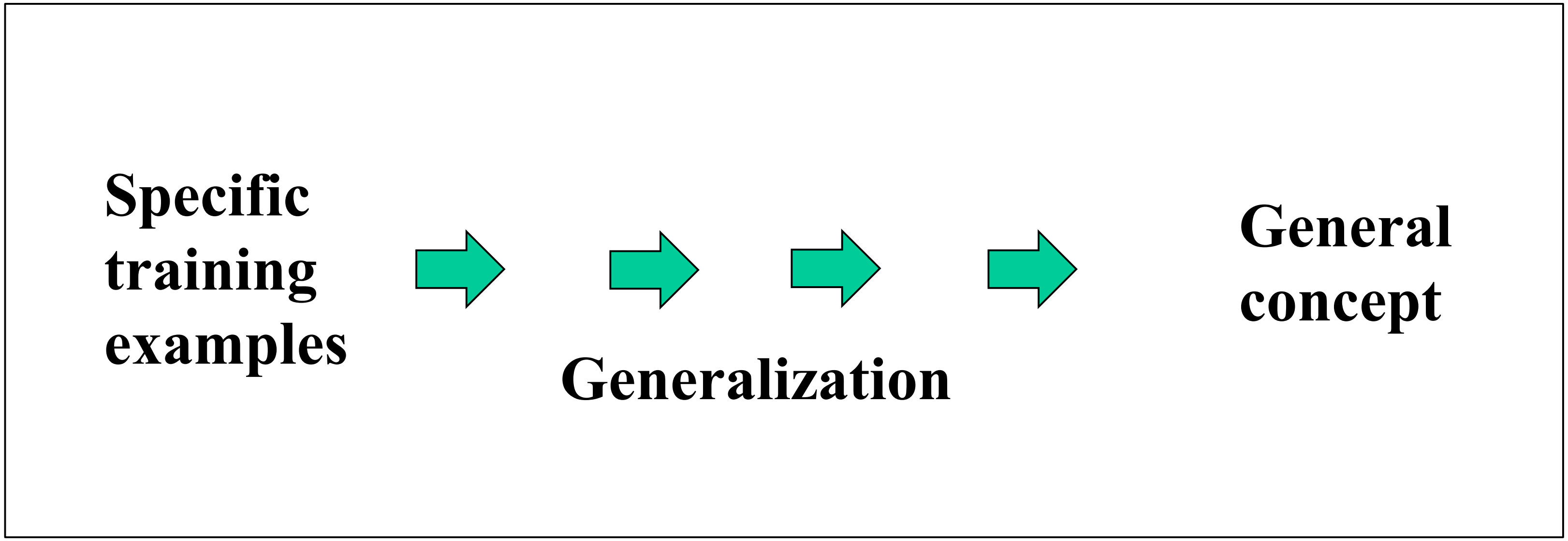
Then h_j is more_general_than h_k ($h_j >_g h_k$)

if and only if

$$(h_j \geq_g h_k) \wedge (h_k \not\geq_g h_j)$$

Concept learning and general to specific ordering

Learning



Find-S: Finding a Maximally Specific Hypothesis

Step 1: $h \leftarrow (\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$

Step 2: $h \leftarrow$ (Sunny, Warm, Normal, Strong, Warm, Same)

Step 3: $h \leftarrow$ (Sunny, Warm, ?, Strong, Warm, Same)

Step 4: $h \leftarrow$ (Sunny, Warm, ?, Strong, ?, ?)

Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

Find-S illustrates how the *more_general_than* partial ordering can be used to organize the search

Find-S: Critical Questions:

1. **Why prefer the most specific hypothesis?**
2. **May other consistent hypotheses be better?**
3. **Is ignoring information about negative examples justified?**
4. **No certificate of training data being inconsistent.**

Version Space and
the Candidate Elimination Algorithm

Main idea: instead of finding one hypothesis consistent with the training examples, the candidate elimination algorithm finds a set of all hypotheses consistent with the training examples.

Version Space and the Candidate Elimination Algorithm

Definitions

A hypothesis h is consistent with a set of training examples D if and only if $h(x) = c(x)$ for each example $(x, c(x))$ in D.

The version space $VS(H,D)$ with respect to hypothesis space H and training examples D is the subset of hypothesis from H consistent with D.

The general (specific) boundary G (S), with respect to hypothesis space H and training examples D is the set of maximally general (maximally specific) members of H consistent with D.

General and specific boundaries

Definition: The **general boundary** G, with respect to hypothesis space H and training data D, is the set of maximally general members of H consistent with D.

$$G \equiv \{g \in H \mid Consistent(g, D) \wedge (\neg \exists g' \in H)[(g' >_g g) \wedge Consistent(g', D)]\}$$

Definition: The **specific boundary** S, with respect to hypothesis space H and training data D, is the set of minimally general (i.e., maximally specific) members of H consistent with D.

$$S \equiv \{s \in H \mid Consistent(s, D) \wedge (\neg \exists s' \in H)[(s >_g s') \wedge Consistent(s', D)]\}$$

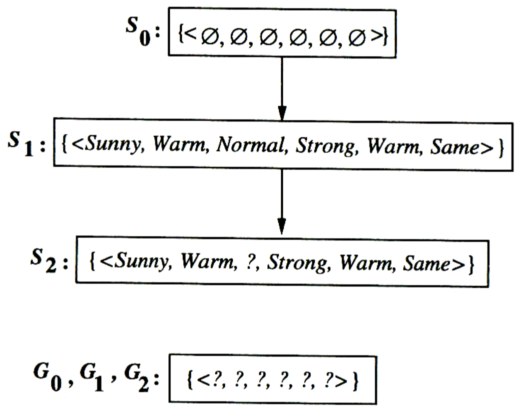
Version Space and the Candidate Elimination Algorithm

Theorem

Let *X* be an arbitrary set of instances and let *H* be a set of boolean-valued hypotheses defined over *X*. Let *c*: *X* → {0, 1} be an arbitrary target concept defined over *X*, and let *D* be an arbitrary set of training examples (*x*, *c*(*x*)), *G* and *S* are general and specific boundaries. Then

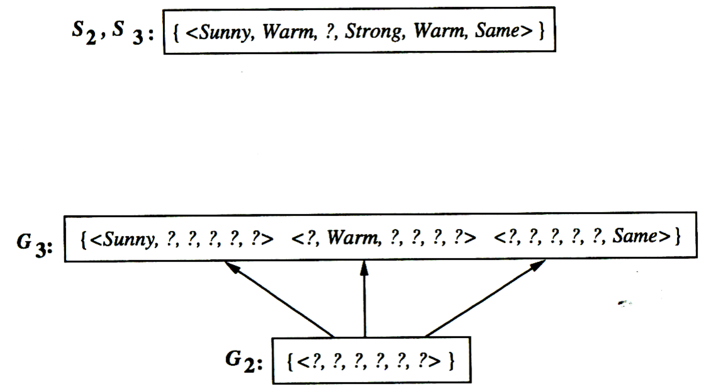
$$VS(H,D) = \left\{ h \in H \mid \exists s \in S,\ g \in G : g \geq_g h \geq_g s \right\}$$

Candidate Elimination Algorithm



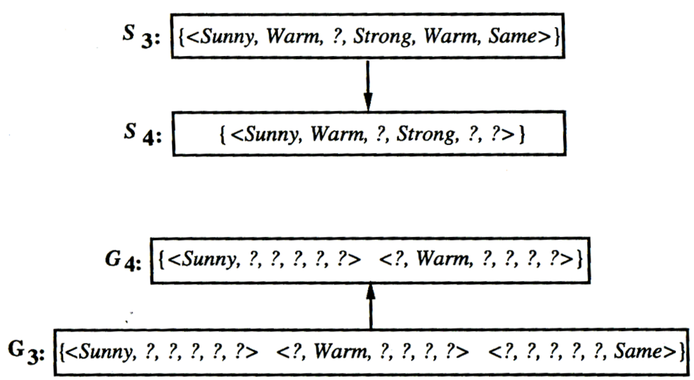
Training examples:

1. *<Sunny, Warm, Normal, Strong, Warm, Same>, Enjoy Sport = Yes*
2. *<Sunny, Warm, High, Strong, Warm, Same>, Enjoy Sport = Yes*



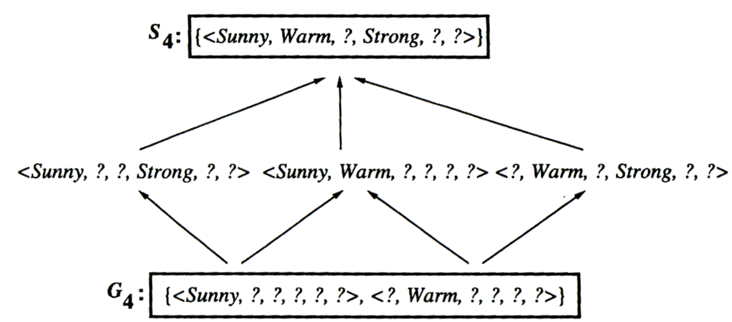
Training Example:

3. *<Rainy, Cold, High, Strong, Warm, Change>, EnjoySport=No*



Training Example:

4. *<Sunny, Warm, High, Strong, Cool, Change>, EnjoySport = Yes*



Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

Candidate Elimination Algorithm (CEA)

Will the CEA converge to the correct target concept?

Yes, if

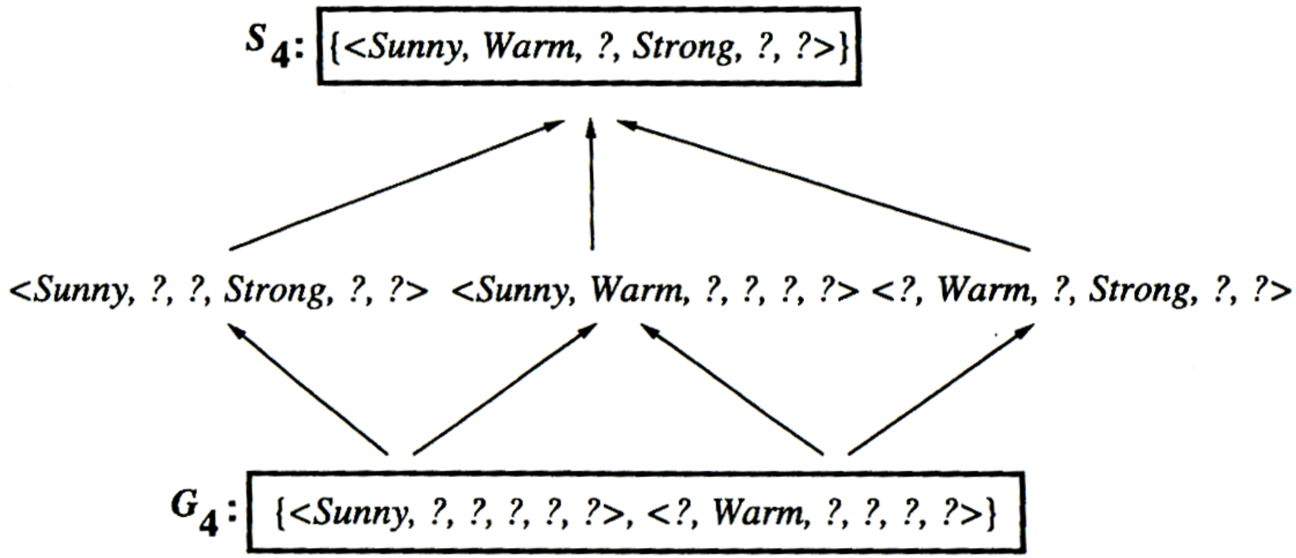
1) there is no errors in the training examples
2) *H* contains the target concept

Convergence occurs when the Version Space become a single hypothesis

What training examples should the learner request next?

Reduce the Version Space at least by half, then one could estimate the number of training examples needed

Large Version Space = Partially Learned Concept



Instance	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
A	Sunny	Warm	Normal	Strong	Cool	Change	?
B	Rainy	Cold	Normal	Light	Warm	Same	?
C	Sunny	Warm	Normal	Light	Warm	Same	?
D	Sunny	Cold	Normal	Strong	Warm	Same	?

Biased Learner

Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Cool	Change	Yes
2	Cloudy	Warm	Normal	Strong	Cool	Change	Yes
3	Rainy	Warm	Normal	Strong	Cool	Change	No

$$S_2 : \langle ?, Warm, Normal, Strong, Cool, Change \rangle$$

Unbiased Learner

$$\langle Sunny, ?, ?, ?, ?, ? \rangle \vee \langle Cloudy, ?, ?, ?, ?, ? \rangle$$

Biased / Unbiased Learner

A learner that make no a priori assumptions on the target concept has no rational basis for classifying any unseen instances!

These assumptions = inductive bias of a learning algorithm!

Inductive bias of learning algorithms

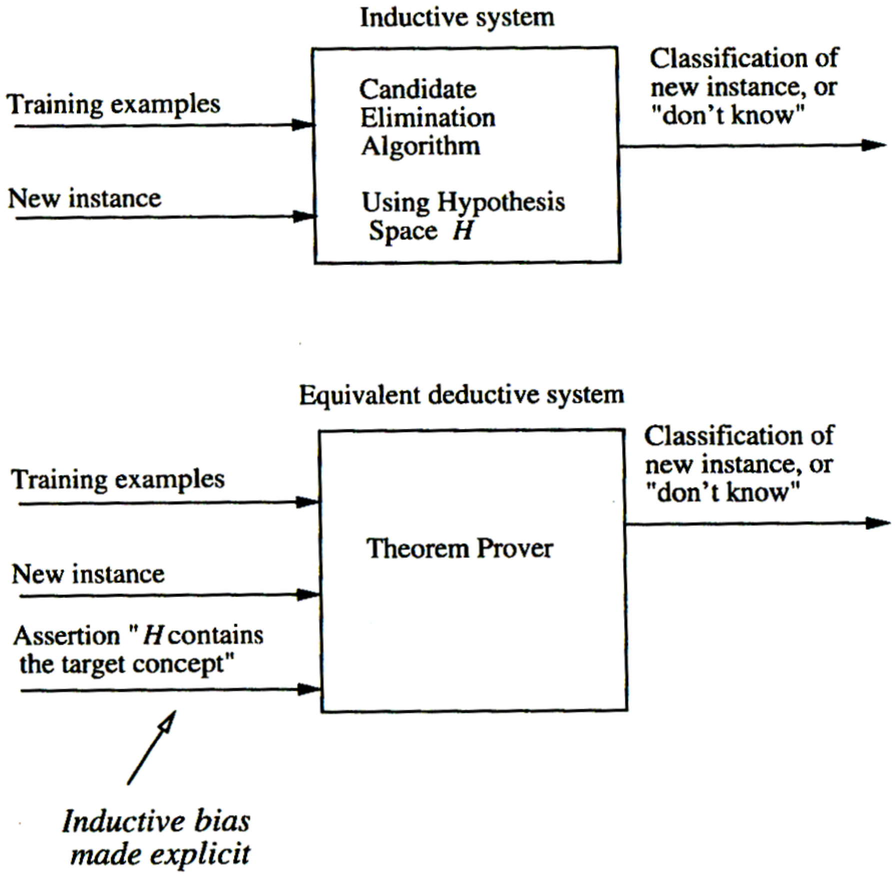
Definition: Consider a concept learning algorithm L for the set of instances X. Let c be an arbitrary concept defined over X, and let $D_c = \{\langle x, c(x)\rangle\}$ be an arbitrary set of training examples of c. Let $L(x_i, D_c)$ denote the classification assigned to the instance x_i by L after training on the data D_c. 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

$$(\forall x_i \in X)[(B \wedge D_c \wedge x_i) \vdash L(x_i, D_c)]$$

Classification can be deducted

Inductive bias of Candidate Elimination algorithm

Inductive bias of CANDIDATE-ELIMINATION algorithm. The target concept c is contained in the given hypothesis space H.



Inductive bias of algorithms

1. **Rote-Leaner.** **There is no inductive bias.**
2. **Candidate Elimination algorithm.** **The target concept is in its hypothesis space.**
3. **Find-S algorithm.** **The target concept is in its hypothesis space + all instances are negative unless the opposite is learned?**

SUMMARY

- Concept learning is a problem of searching through a large predefined space of potential hypotheses

- The general-to-specific partial ordering provides a useful structure for organizing the search through the hypothesis space

- The Find-S algorithm finds the most specific hypothesis consistent with the training examples

- The Candidate Elimination algorithm computes the version space by incrementally computing the sets of maximally specific and maximally general hypotheses

- The Candidate elimination algorithm is not robust to noisy data, or to situations in which the target concept is not expressible in provided hypothesis space

- Inductive learning algorithms are able to generalize because of their implicit inductive bias.

- Unbiased learner cannot generalize!