Module 12

Machine Learning

Version 1 CSE IIT, Kharagpur

Lesson 34

Learning From Observations

Version 1 CSE IIT, Kharagpur

12.2 Concept Learning

Definition:

The problem is to learn a function mapping examples into two classes: positive and negative. We are given a database of examples already classified as positive or negative. Concept learning: the process of inducing a function mapping input examples into a Boolean output.

Examples:

- Classifying objects in astronomical images as stars or galaxies
- Classifying animals as vertebrates or invertebrates

Example: Classifying Mushrooms

Class of Tasks: Predicting poisonous mushrooms
Performance: Accuracy of classification
Experience: Database describing mushrooms with their class

Knowledge to learn: Function mapping mushrooms to {0,1} where 0:not-poisonous and 1:poisonous

Representation of target knowledge: conjunction of attribute values.

Learning mechanism: candidate-elimination

Representation of instances:

Features:

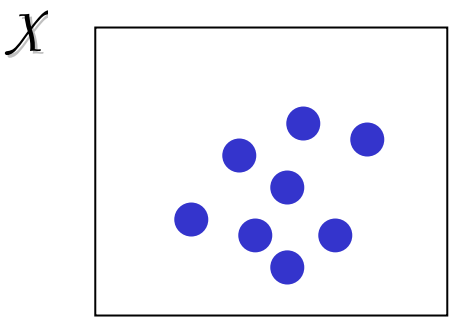
- color {red, brown, gray}
- size {small, large}
- shape {round,elongated}
- land {humid,dry}
- air humidity {low,high}
- texture {smooth, rough}

Input and Output Spaces:

X : The space of all possible examples (input space).
Y: The space of classes (output space).

An example in X is a feature vector X.
For instance: X = (red,small,elongated,humid,low,rough)
X is the cross product of all feature values.

Only a small subset of instances is available in the database of examples.



Y= {0,1}

Training Examples:

D : The set of training examples.
D is a set of pairs { (x,c(x)) }, where c is the target concept. c is a subset of the universe of discourse or the set of all possible instances.

Example of D:

((red,small,round,humid,low,smooth),	poisonous)
((red,small,elongated,humid,low,smooth),	poisonous)
((gray,large,elongated,humid,low,rough),	not-poisonous)
((red,small,elongated,humid,high,rough),	poisonous)

Hypothesis Representation

Any hypothesis h is a function from X to Y
h: X → Y

We will explore the space of conjunctions.

Special symbols:

- ? Any value is acceptable
- 0 no value is acceptable

Consider the following hypotheses:
(?,?,?,?,?,?): all mushrooms are poisonous
(0,0,0,0,0,0): no mushroom is poisonous

Hypotheses Space:

The space of all hypotheses is represented by H

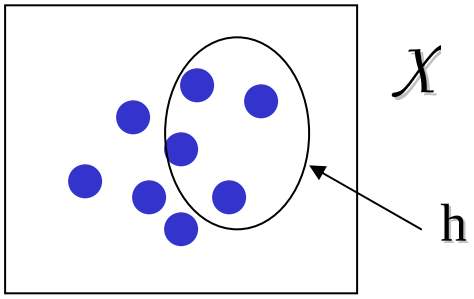
Let h be a hypothesis in H.

Let X be an example of a mushroom.

if h(X) = 1 then X is poisonous, otherwise X is not-poisonous

Our goal is to find the hypothesis, h*, that is very "close" to target concept c.

A hypothesis is said to "cover" those examples it classifies as positive.



Assumptions:

We will explore the space of all conjunctions.
We assume the target concept falls within this space.

A hypothesis close to target concept c obtained after seeing many training examples will result in high accuracy on the set of unobserved examples. (Inductive Learning Hypothesis)

12.2.1 Concept Learning as Search

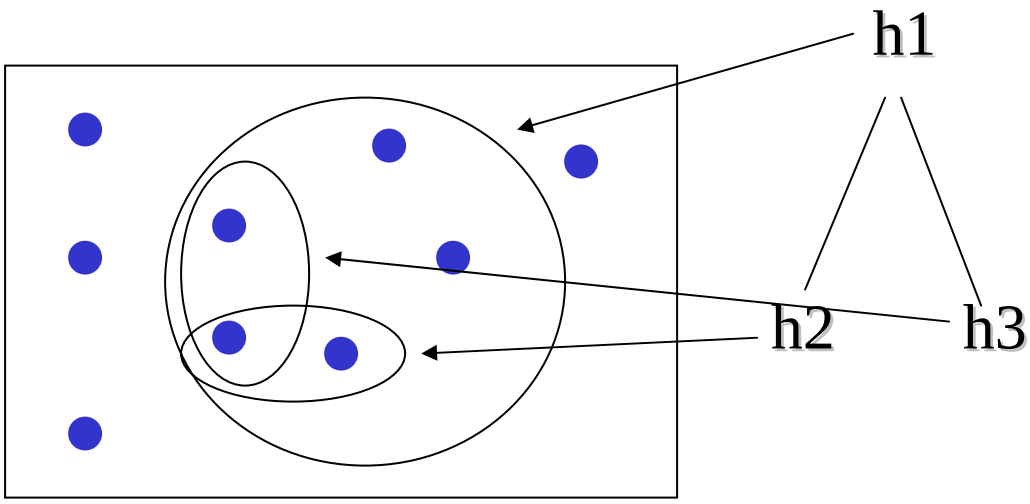
We will see how the problem of concept learning can be posed as a search problem.

We will illustrate that there is a general to specific ordering inherent to any hypothesis space.

Consider these two hypotheses:
h1 = (red,?,?,humid,?,?)
h2 = (red,?,?,?,?,?)

We say h2 is more general than h1 because h2 classifies more instances than h1 and h1 is covered by h2.

For example, consider the following hypotheses



h1 is more general than h2 and h3.
h2 and h3 are neither more specific nor more general than each other.

Definitions:

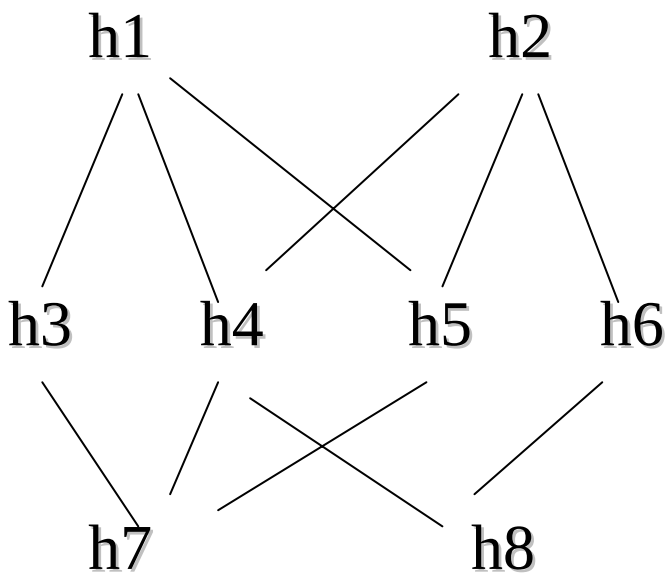
Let **hj** and **hk** be two hypotheses mapping examples into {0,1}.
We say **hj** is more general than **hk** iff

For all examples X, hk(X) = 1 → hj(X) = 1

We represent this fact as **hj** >= **hk**

The >= relation imposes a partial ordering over the hypothesis space **H** (reflexive, antisymmetric, and transitive).

Any input space X defines then a lattice of hypotheses ordered according to the general-specific relation:



12.2.1.1 Algorithm to Find a Maximally-Specific Hypothesis

Algorithm to search the space of conjunctions:

- Start with the most specific hypothesis
- Generalize the hypothesis when it fails to cover a positive example

Algorithm:

1. Initialize **h** to the most specific hypothesis
2. For each positive training example **X**
   For each value **a** in **h**
   If example **X** and **h** agree on **a,** do nothing
   Else generalize **a** by the next more general constraint
3. Output hypothesis **h**

Example:

Let's run the learning algorithm above with the following examples:

((red,small,round,humid,low,smooth),	poisonous)
((red,small,elongated,humid,low,smooth),	poisonous)
((gray,large,elongated,humid,low,rough),	not-poisonous)
((red,small,elongated,humid,high,rough),	poisonous)

We start with the most specific hypothesis: **h** = (0,0,0,0,0,0)

The first example comes and since the example is positive and **h** fails to cover it, we simply generalize **h** to cover exactly this example:
h = (red,small,round,humid,low,smooth)

Hypothesis **h** basically says that the first example is the only positive example, all other examples are negative.

Then comes examples 2: ((red,small,elongated,humid,low,smooth), poisonous)

This example is positive. All attributes match hypothesis **h** except for attribute shape: it has the value *elongated*, not *round*. We generalize this attribute using symbol ? yielding:

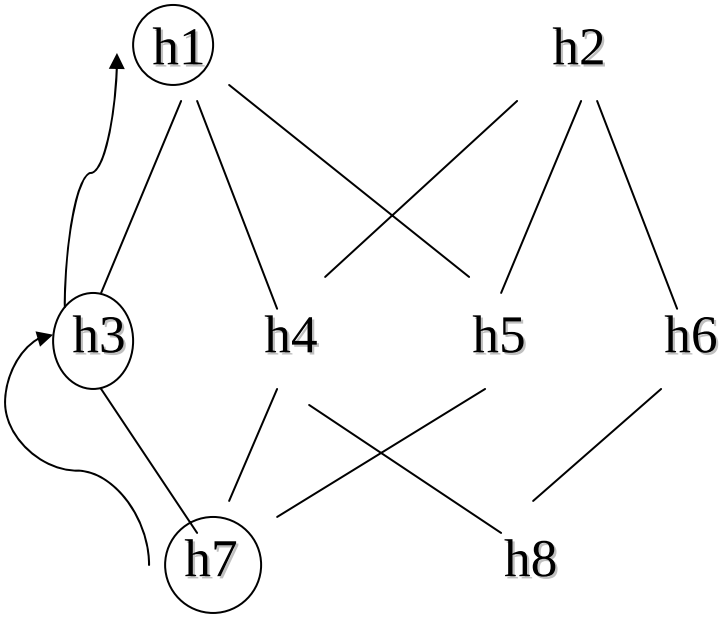
h: (red,small,?,humid,low,smooth)
The third example is negative and so we just ignore it.

Why is it we don't need to be concerned with negative examples?

Upon observing the 4th example, hypothesis **h** is generalized to the following:

h = (red,small,?,humid,?,?)

h is interpreted as any mushroom that is red, small and found on humid land should be classified as poisonous.



The algorithm is guaranteed to find the hypothesis that is most specific and consistent with the set of training examples. It takes advantage of the general-specific ordering to move on the corresponding lattice searching for the next most specific hypothesis.

Note that:

There are many hypotheses consistent with the training data D. Why should we prefer the most specific hypothesis?

What would happen if the examples are not consistent? What would happen if they have errors, noise?

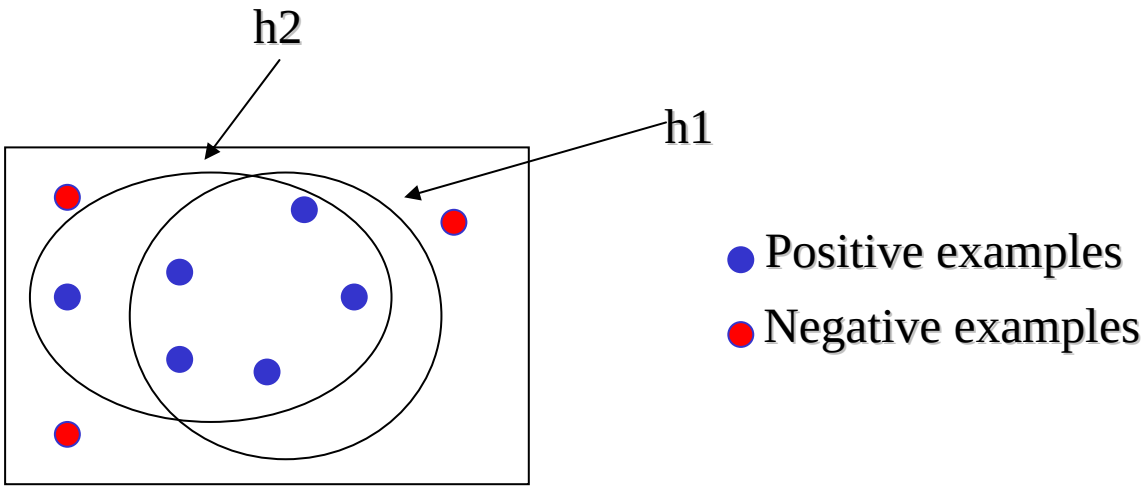
What if there is a hypothesis space H where one can find more that one maximally specific hypothesis h? The search over the lattice must then be different to allow for this possibility.

- The algorithm that finds the maximally specific hypothesis is limited in that it only finds one of many hypotheses consistent with the training data.
- The Candidate Elimination Algorithm (CEA) finds ALL hypotheses consistent with the training data.
- CEA does that without explicitly enumerating all consistent hypotheses.
- Applications:
    - Chemical Mass Spectroscopy
    - Control Rules for Heuristic Search

12.2.2 Candidate Elimination Algorithm

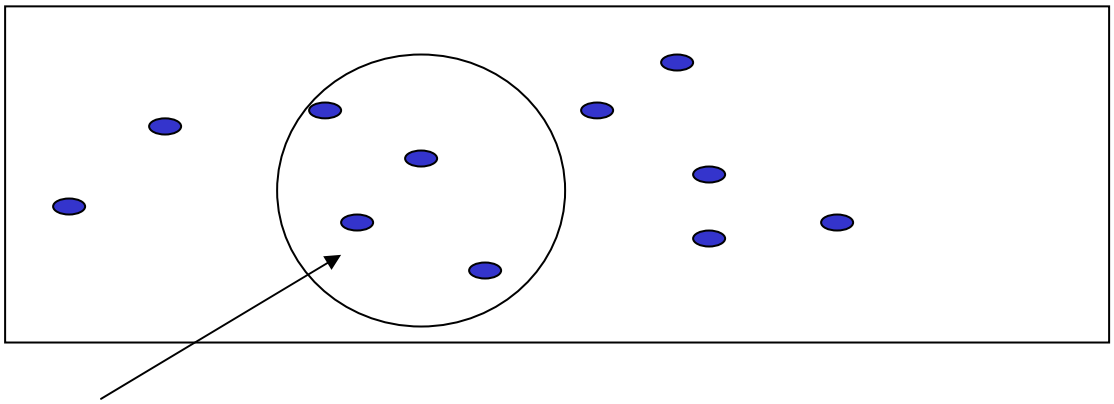
Consistency vs Coverage

In the following example, h1 covers a different set of examples than h2, h2 is consistent with training set D, h1 is not consistent with training set D



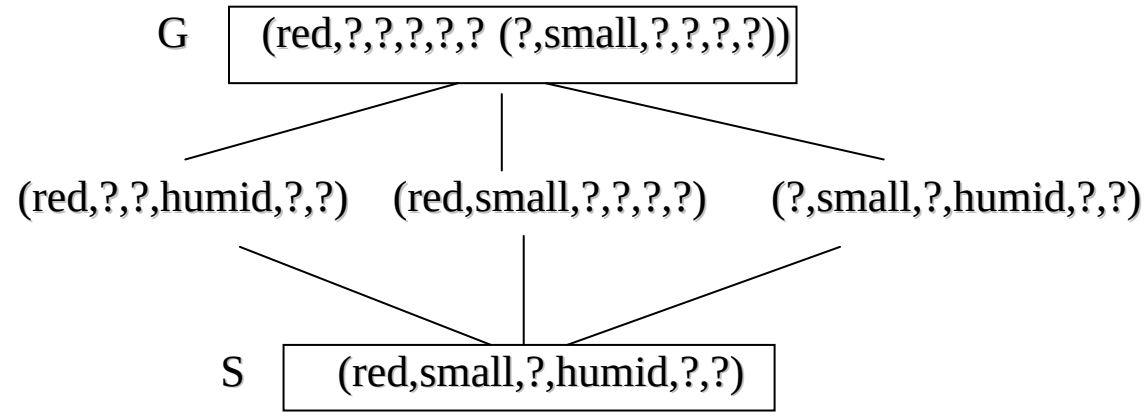
12.2.2.1 Version Space

Hypothesis space $\mathcal{H}$



Version space: Subset of hypothesis from $\mathcal{H}$ consistent with training set $\mathcal{D}$.

The version space for the mushroom example is as follows:



S: Most specific
G: Most general

The candidate elimination algorithm generates the entire version space.

12.2.2.2 The Candidate-Elimination Algorithm

The candidate elimination algorithm keeps two lists of hypotheses consistent with the training data: (i) The list of most specific hypotheses **S** and, (ii) The list of most general hypotheses **G.** This is enough to derive the whole version space **VS**.

Steps:

1. Initialize G to the set of maximally general hypotheses in H
2. Initialize S to the set of maximally specific hypotheses in H
3. For each training example X do
    a) If X is positive: generalize S if necessary
    b) If X is negative: specialize G if necessary
4. Output {G,S}

Step (a) Positive examples

If X is positive:

- Remove from G any hypothesis inconsistent with X
- For each hypothesis h in S not consistent with X
    - Remove h from S
    - Add all minimal generalizations of h consistent with X such that some member of G is more general than h
    - Remove from S any hypothesis more general than any other hypothesis in S

Step (b) Negative examples

If X is negative:

- Remove from S any hypothesis inconsistent with X
- For each hypothesis h in G not consistent with X
    - Remove g from G
    - Add all minimal generalizations of h consistent with X such that some member of S is more specific than h
    - Remove from G any hypothesis less general than any other hypothesis in G

The candidate elimination algorithm is guaranteed to converge to the right hypothesis provided the following:

a) No errors exist in the examples
b) The target concept is included in the hypothesis space **H**

If there exists errors in the examples:

a) The right hypothesis would be inconsistent and thus eliminated.
b) If the **S** and **G** sets converge to an empty space we have evidence that the true concept lies outside space **H**.