Concept Learning

- concept

  some subset of objects or events defined over a large set

  example.

  the subset of animals that constitute birds

  representation of concept:

  a boolean valued function defined over a large set

  example.

  a function defined over all animals, whose value is true (1) for birds and false (0) for other animals

- learning

  inducing general functions from specific training examples

- concept learning (or category learning)

  acquiring the definition of general category given a sample of positive and negative training examples of category, that is, inferring a boolean-valued function from training examples of its input and output

- a concept learning task
  - . target concept: EnjoySport
    (days on which Aldo enjoys water sport)
  - . hypothesis: a vector of six constraints,
    specifying the value of six attributes, they are,
    Sky (Sunny/Cloudy/Rainy), AirTemp (Warm/Cold),
    Humidity (Normal/High), Wind (Strong/Weak),
    Water (Warm/Cool), Forecast (Same/Change)
    for each attribute, the hypothesis will either
    ? (don't care: any value is acceptable),
    discrete values, or
    ∅ (null: no value is acceptable)

example.

<?, Cold, High, ?, ?, ?>
-> "Aldo enjoys sport only on cold days with high humidity."

<?, ?, ?, ?, ?, ?>
-> "Aldo always enjoys sport." (most general hypothesis)

<∅, ∅, ∅, ∅, ∅, ∅>
-> "Aldo does not enjoy sport at all." (most specific case)

. Positive and negative training examples for the target concept 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

What is *the general concept* for these examples?

Given

instances X: possible days, each described by six attribute,

target function c: EnjoySport $X \rightarrow \{0, 1\}$,

hypothesis H: conjunction of literals such as

<?, Cold, High, ?, ?, ?>, and

training examples D: positive and negative examples of the target function, that is,

$< X_1, c(X_1) >, \cdots, < X_m, c(X_m) >$,

determine

a hypothesis h in H such that

$h(x) = c(x)$ for all x in X.

- inductive learning hypothesis

Any hypothesis found to be approximate the target function well over *a sufficiently large set training examples* will also approximate the target function well over *other unobserved examples*.

- concept learning as search

find a hypothesis that best fits training examples

search space in EnjoySport:

number of instances = $3 \cdot 2^5 = 96$

number of hypotheses = $5 \cdot 4^5 = 5120$

- general-to-specific ordering

. Let $x \in X$ and $h \in H$. Then,

x satisfies h if and only if $h(x) = 1$.

. 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 ($h_j \geqq_g h_k$) if and only if

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

h_j is *(strictly) more_general_than* h_k ($h_j >_g h_k$) if and only if

$(h_j \geqq_g h_k) \wedge \neg (h_k \geqq_g h_j)$.

example.

h_j=<Sunny, ?, ?, ?, ?, ?> $>_g$ h_k=<Sunny, ?, ?, Strong, ?, ?>

$\rightarrow$ h_j is more_general_than h_k. or

$\rightarrow$ h_k is more_specific_than h_j.

Here, the problem is how to search the good hypothesis using this hypothesis ordering.

One of such candidates is Find-S algorithm in which the maximally specific hypothesis is searched.

- Find-S algorithm

Step 1. Initialize h to the most specific hypothesis in H, that is,

h = <∅, ∅, ∅, ∅, ∅, ∅>.

Step 2. For each ***positive training instance*** x

- for each attribute constraint a_i in h

  if the constraint a_i in h is satisfied by x, do nothing

  else replace a_i in h by the next more general constraint that is satisfied by x.

Step 3. Output h.

example.

h_0=<∅, ∅, ∅, ∅, ∅, ∅>.

x_1=<Sunny, Warm, Normal, Strong, Warm, Same>+

h_1=<Sunny, Warm, Normal, Strong, Warm, Same>

x_2=<Sunny, Warm, High, Strong, Warm, Same>+

h_2=<Sunny, Warm, ?, Strong, Warm, Same>

x_3=<Rainy, Cold, High, Strong, Warm, Change>-

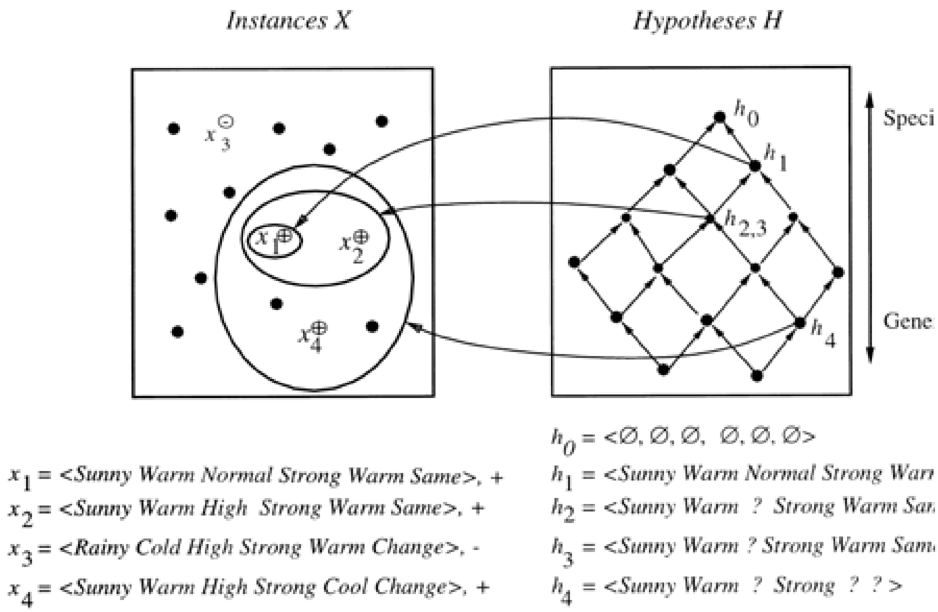
h_3=h_2

x_4=<Sunny, Warm, High, Strong, Cool, Change>+

h_4=<Sunny, Warm, ?, Strong, ?, ?>

… … …

Hypothesis space searched by Find-S algorithm



- problem in Find-S algorithm

  . can't tell whether it has the learned concept.

  . can't tell when training data are inconsistent.

  . picks a maximally specific h.

  . depending on H, there might be several.

-> Find-S algorithm only uses the positive examples.

-> We better find *the proper hypothesis space* rather than a specific hypothesis.

-> the concept of version spaces

Reference: Machine Learning, chapter 2.