Concept Learning

- Learning Concepts from Examples
- "Concept" typically means categorization based on features

A Concept Learning Task

Four Examples:

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

- Target concept:
 - "Days on which Aldo enjoys his favorite water sport"
 - Two categories: yes/no

A Concept Learning Task

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

- Some possible concepts:
 - Enjoy sport when Sky = Sunny
 - Enjoy sport when Airtemp = Warm

Concept Learning Methodology

- Inducing general functions from specific training examples
 - Concerns acquiring general concepts (or categories) from training examples based on symbolic or logical representations
 - Formulated as:
 - search through space of potential hypotheses

What is a concept?

- Concepts:
 - "bird", "car", "situations in which I should study more in order to pass the exam"
- Each concept describes a subset of objects
 - subset of animals that constitute birds
 - Each concept is a boolean-valued function defined over this larger set
 - function defined over all animals whose value is true for birds and false for other animals

Concept Learning

 Automatically inferring the general definition of some concept, given examples labeled as members or non-members of the concept

 Concept Learning: Inferring a boolean-valued function from training examples of its input and output.

A Concept Learning Task

- Target concept:
 - "Days on which Aldo enjoys his favorite water sport"

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

- Task is to learn to predict value of EnjoySport
 - for an arbitrary day based on values of other attributes

Hypothesis Representation by Learner

- Conjunction of constraints on attributes
 - Sky, AirTemp, Humidity, Wind, Water, Forecast
- For each attribute
 - indicate by a "?" that any value is acceptable for this attribute
 - specify a single required value for the attribute (eg, Warm)
 - indicate by a Φ that no value is acceptable

Three Example Hypotheses

- H1: Enjoy sport only on cold days with high humidity
 - (?,Cold,High,?,?,?)
- H2: Most general hypothesis
 - every day is a positive example
 - **(?,?,?,?,?)**
- H3: Most specific possible hypothesis
 - no day is a positive example
 - (Ф,Ф,Ф,Ф,Ф)

Hypothesis Representation

- If x is an instance that satisfies all constraints of hypothesis h, then h classifies x as a positive example ((h(x)=1)
- Each hypothesis defines a particular definition of a categorizer

Notation

- Set of items over which concept is defined: instances, denoted by X
 - X is set of all possible days, each represented by attributes,
 Sky, AirTemp, Humidity, Wind, Water and Forecast
- Concept or function to be learned: target concept, c
 - c can be any boolean-valued function defined over the instances
 - **c**: X -->{0,1}
- Learner is presented a set of Training Examples
 - c(x)=1 are positive examples, c(x)=0 are negative examples
- Set of all possible hypotheses, H
 - each hypothesis h in H is a function h: X -->{0,1}
- Goal of Learner is to find hypothesis h, such that h(x)=c(x) for all x in X

EnjoySport Concept Learning Task

Given

- Instances X: Possible days, described by attributes
 - Sky (with possible values Sunny, Cloudy, and Rainy)
 - AirTemp (with values Warm and Cold)
 - Humidity (with values Normal and High)
 - Wind (with values Strong and Weak)
 - Water (with values Warm and Cool)
 - Forecast (with values Same and Change)
- Hypotheses H: Each hypothesis is described by a conjunction of the attributes Sky, AirTemp,
 Humidity, Wind, Water, and Forecast. The constraints may be "?" (any value is acceptable),
 "Ø" (no value is acceptable), or a specific value
- Target concept $c: EnjoySport: X \rightarrow \{0,1\}$
- Training examples *D*: Positive and negative examples of target function

• Determine

• A hypothesis h such that h(x) = c(x) for all x in X

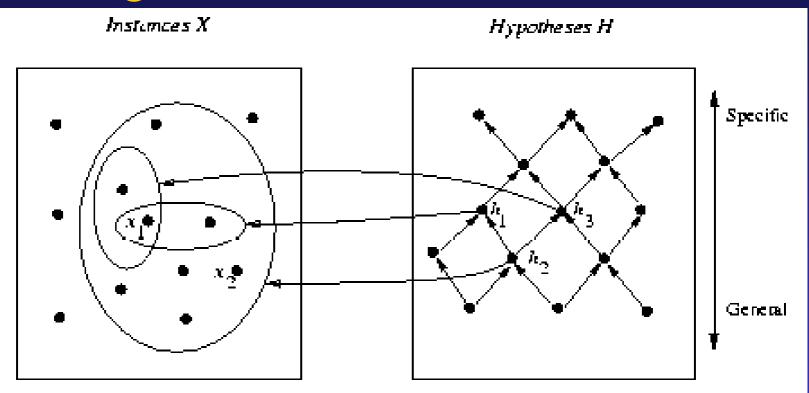
Inductive Learning Hypothesis

- Although the Learning Task is to determine identical to the target concept c over the entire set of instances X, the only information available about c is its value over the training examples
- 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 samples

Concept Learning As Search

- Concept learning is a search through a large space of hypotheses (implictly defined by the hypothesis representation)
- Goal is to find hypothesis that best fits training examples

Instances, Hypotheses and more_general_than relation



$$x_1 = \langle Simmy, Wattm, High, Situag, Cool, Same >$$

 $x_2 = \langle Simmy, Wattm, High, Light, Wattm, Same >$

$$\begin{split} h_1 &= \\ h_2 &= \\ h_3 &= \end{split}$$

Size of Instance/Hypotheses Spaces

- Enjoysport learning Task
 - sky: 3 values, rest: 2 values
 - instance space X: 3.2.2.2.2=96 instances
 - hypotheses:
 - 5.4.4.4.4 = 5120 syntactically distinct hypotheses
 - every hypothesis containing Ø classifies instance as negative
 - $\mathbf{1}$ + (4.3.3.3.3.3) = 973 semantically distinct hypotheses
 - since null values classify every instance as negative
 - extra one for all null values

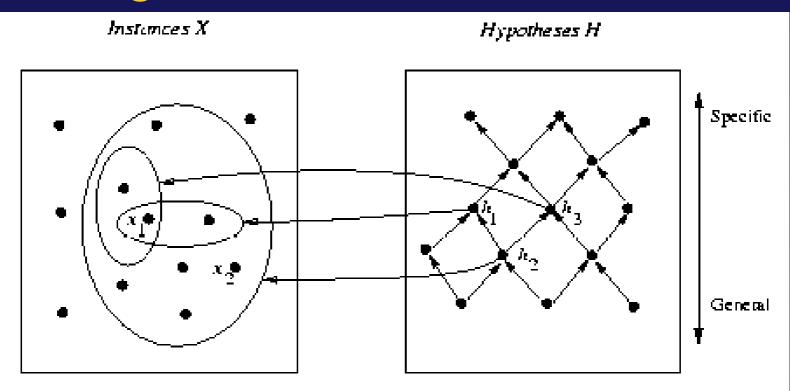
General to Specific Ordering of Hypotheses

- Algorithm can search through hypothesis space by
 - taking advantage of ordering of hypotheses that exists
 - not enumerating every possible hypothesis
- $h_1 = \langle Sunny, ?, ?, Strong, ?, ? \rangle$
- $h_2 = \langle Sunny, ?, ?, ?, ?, ? \rangle$
- Since h₂ imposes fewer contraints on the instance, it classifies more instances as positive
- h₂ is more general than h₁

More_general_than_ or_equal_to_relation

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 (written $h_j \ge h_k$) if and only if $(\forall_x \in X)[(h_k(x)=1) \to (h_j(x)=1)]$

Instances, Hypotheses and more_general_than relation



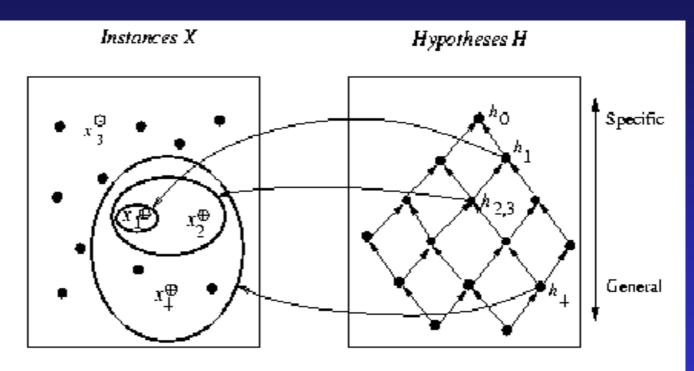
x₁= <Simny, Wairin, High, Strong, Cool, Same> x₂= <Simny, Wairin, High, Light, Wairin, Saine>

$$\begin{aligned} h_1 &= \\ h_2 &= \\ h_3 &= \end{aligned}$$

FIND-S Algorithm

- Initialize h to the most specific hypothesis in
- For each positive training instance x
 - For each attribute constraint a; in h
 - If the constraint a_i is satisfied by x
 - Then do nothing
 - Else replace $\mathbf{a_i}$ in h by the next more general constraint that is satisfied by \mathbf{x}
- Output hypothesis h

Hypothesis Space Search Performed by FIND-S



+:positive instance -: negative instance

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

$$h_0 = <\varnothing, \varnothing, \varnothing, \varnothing, \varnothing, \varnothing, \varnothing>$$
 $h_1 =
 $h_2 =$
 $h_3 =$
 $h_4 =$$

FIND-S: Finding a Maximally Specific Hypothesis

- Step1: h <-- <0,0,0,0,0,0,0</p>
- Upon observing first training example, current hypothesis is too specific. So replace by the next more general constraint
 - h <-- <Sunny,Warm,Normal,Strong,Warm,Same>
- After observing second training sample
 - h <--<Sunny,Warm,?,Strong,Warm, Same>
- Ignore third sample (which is negative)
- After fourth sample
 - h <--<Sunny,Warm,?,Strong,?>

Hypothesis found by FIND-S

h <--<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 Algorithm Properties

- Uses more_general_than partial ordering to organize search for an acceptable hypothesis
- Search moves from hyp to hyp searching from most specific to more general hyps
- At each step hyp is generalized only as far as necessary
- Guaranteed to output most specific hyp consistent with positive training examples

Find-S has limitations

- Has the learner converged to the target concept?
 - Would like to characterize uncertainty in learning true identity of target concept
- Why prefer the most specific hypothesis?
 - Some intermediate generality may be preferable
- Are the training examples consistent?
 - Error or noise in training data can severely mislead Find-S
- What if there are several maximally specific hypotheses?
 - For other hypothesis spaces there can be several