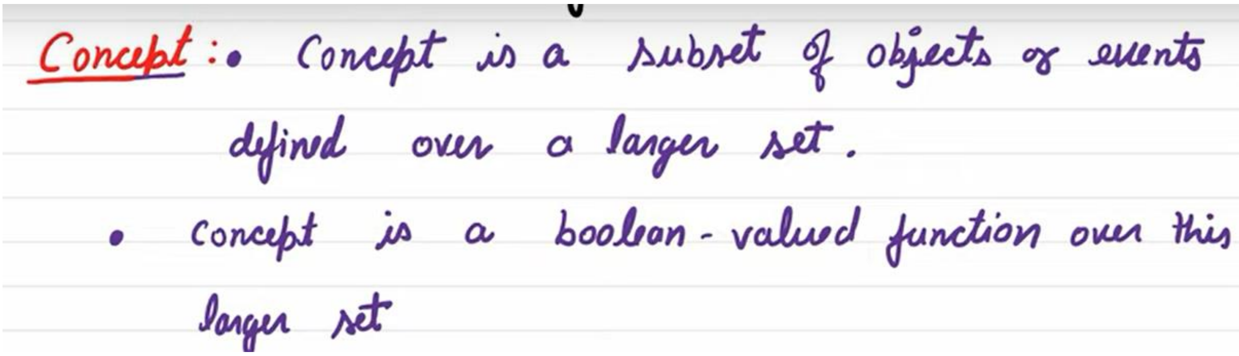
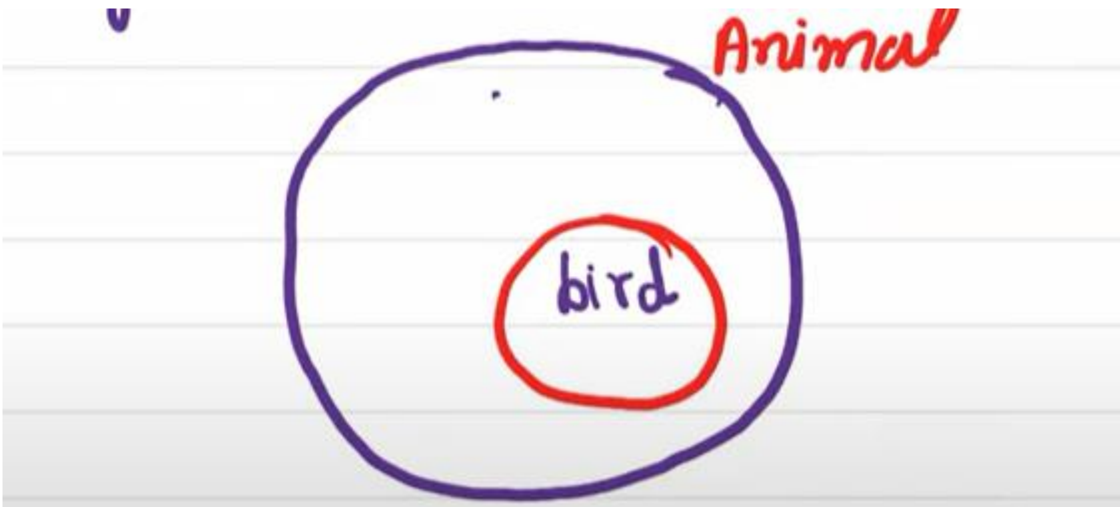
CONCEPT LEARNING



Concept: • Concept is a subset of objects or events defined over a larger set.

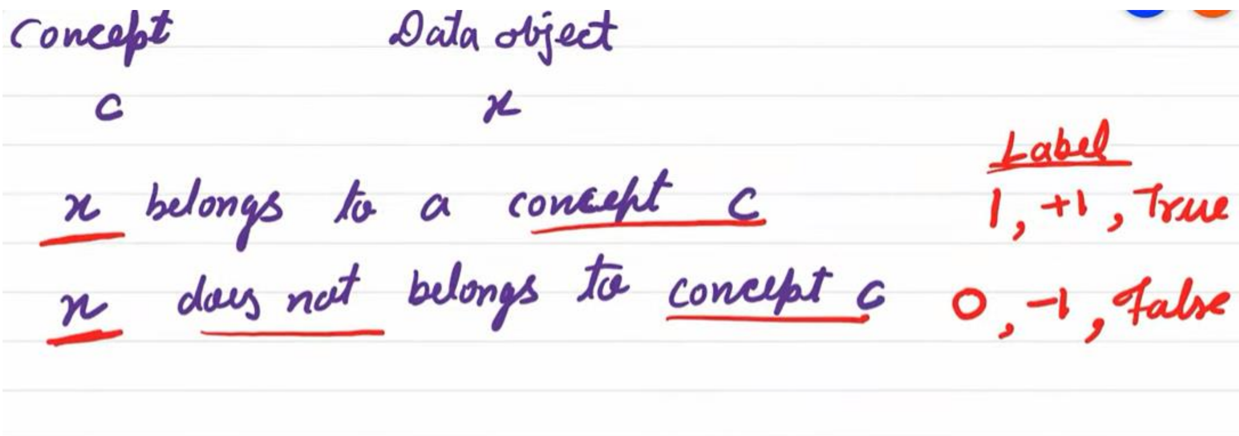
• Concept is a boolean-valued function over this larger set



Animal – larger dataset

Bird-subset

<u>Boolean valued function</u>- Whether the animal is member of the bird or not



What concept learning will do?

- **Concept learning** is the task of automatically inferring the general definition of some concept, given examples labelled as members or non members of the concept.
- **Concept learning -> Finding the best hypothesis**
- **Concept learning is a learning task in which train our machine to learn some concept by giving predefined examples.(training data)**

- Example: Task of learning the target concept “**Days on which my friend X enjoys his favorite water sport"**

  Positive and negative training examples for the target concept “***EnjoySport”***

- The attribute ***EnjoySport*** indicates whether or not Aldo enjoys his favorite water sport on this day.

- **The task is to learn to predict the value of *EnjoySport* for an arbitrary day, based on the values of its other attributes.**

Example of concept learning task.

Concept : Good days for water sports (value : yes, No)

Every concept has certain attributes

Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

According to the dataset, a person enjoying the sports for some days or not enjoying sports some days.

Attributes or Feature:

We can represent like this: (Conjunction of all the attributes)-

< Sunny , warm , High , strong , warm , same , yes >

Consists of input and output variable

Day	Sky	Airtemp	Humidity	Wind	Water	Forecast	Watersport
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

Positive and negative training example for the target concept Enjoysport.

Concept-Enjoysport- yes/no

- Given :
  - instances (*X*): set of items over which the concept is defined.
  - target concept (c) : c : X → {0, 1}
  - training examples (positive/negative) : <x,c(x)>
  - training set D: available training examples
  - set of all possible hypotheses: H
- Determine :
  - to find h(x) = c(x) (for all x in X)

Hypothesis Representation

Hypothesis Representation

For each attribute, the hypothesis will be conjunction of constraints,

sunny ∧ Warm ∧ High ∧ strong cool

Goal : To infer the best concept-description from the set of all possible hypotheses, which can generalise all known or unknown elements of the instance space.

- Hyothesis : h, a conjunction of constraints on the instance attributes.

- Let each hypothesis be a vector of six constraints, specifying the values of the six attributes (***Sky, AirTemp, Humidity, Wind, Water,*** and ***Forecast).***

Instance space-training samples

- Each constraint can be

- ? ϕ

? indicates any value is accepted for attribute

ϕ indicates that no value is accepted.

Specify the single required value of attribute

eg Warm

Six attributes in the dataset can be represented like this:

example: < ? , cold , high , ? ? ? ? >

There are 2 types of hypothesis

① Most General Hypothesis →

$< ?\ ?\ ?\ ?\ ?\ ? >$

② Most specific Hypothesis

$< \phi\ \phi\ \phi\ \phi\ \phi\ \phi >$

Most general hypothesis- All the days are good day for sports

Most specific hypothesis - No day is good day for sports

FIND-S Algorithm

Finding A Maximally Specific Hypothesis

We will use hypothesis representation in Find S algorithm

x_1 < sunny, warm, normal, strong, warm, same> yes

x_2 < Sunny, warm, High, strong, warm, same> yes

x_3 < Rainy, cold, High, strong, warm, change> No

x_4 < Sunny, warm, High, strong, cool, change> yes

6 attributes

Target variable or target concept- EnjoySport- Yes/No (Binary classification)

Sky has two possibilities – sunny or Rainy

AirTemp has possibilities- Warm or cold ….

Given the dataset, a person – enjoying sports some days / not enjoying sports some days

Task - EnjoySport

FIND-S: Step-1

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

Six attributes are there. Initially all the 6 attributes are null.

1. Initialize h to the most specific hypothesis in H

$$h0 = <Ø, Ø, Ø, Ø, Ø, Ø>$$

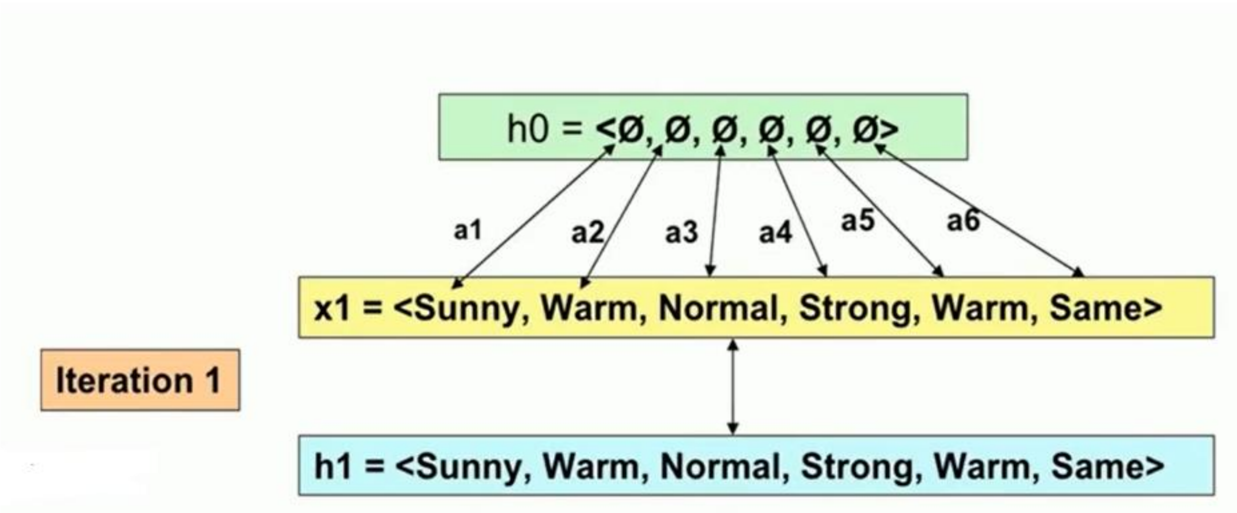
FIND-S: Step-2

Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySpor*
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

2. For each positive training instance x
- For each attribute constraint a_i in h
  If the constraint a_i is satisfied by x
  Then do nothing
  Else replace a_i in h by the next more general constraint that is satisfied by x

[Checking with positive instance x]

Replace with the next general constraint



FIND-S: Step-2

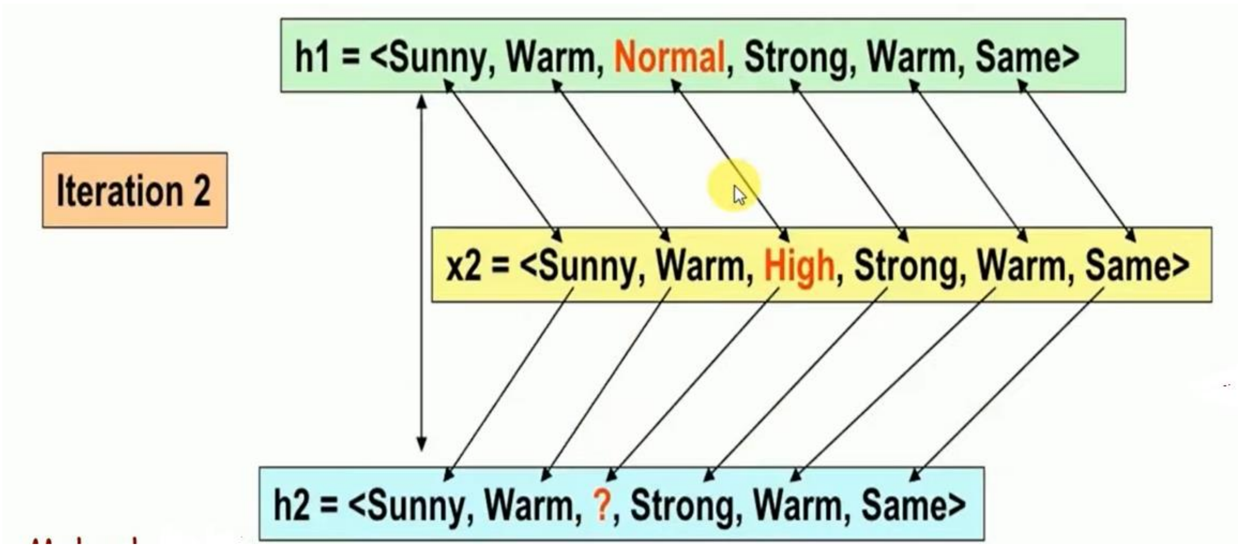
Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySpor*
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

2. For each positive training instance x
   - For each attribute constraint a_i in h

     If the constraint a_i is satisfied by x

     Then do nothing

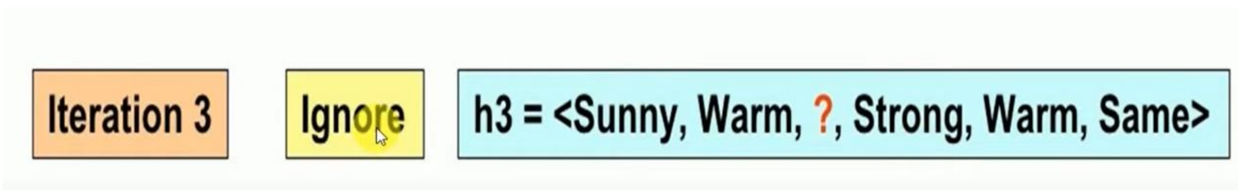
     Else replace a_i in h by the next more general constraint that is satisfied by x



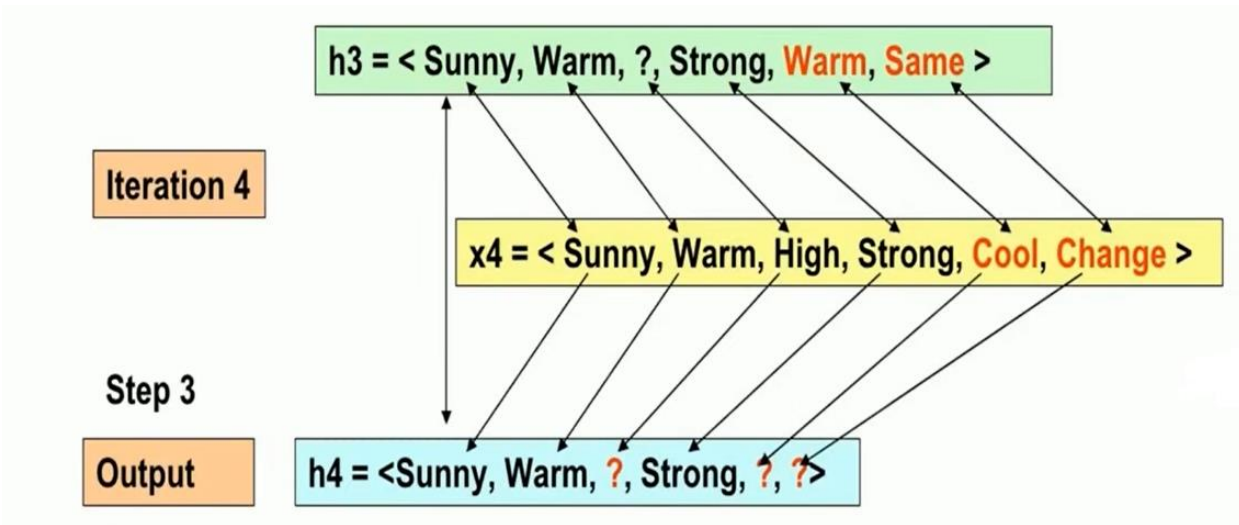
FIND-S: Step-2

Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySpor*
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

2. For each positive training instance x
   - For each attribute constraint a_i in h
     If the constraint a_i is satisfied by x
     Then do nothing
     Else replace a_i in h by the next more general constraint that is satisfied by x



We ignore the training sample 3, because it contains negative training instance



[Checking with next positive instance]

Since the value of 5th attribute changes from ‘Warm’ to ‘Cool’ and 6th attribute changes from ‘same’ to ‘Change’, replace them with [‘?’].

For the given dataset, this is the maximally specific hypothesis.

We have traced all the instances. So this is the final hypothesis.(ie.maximally specific hypothesis.)

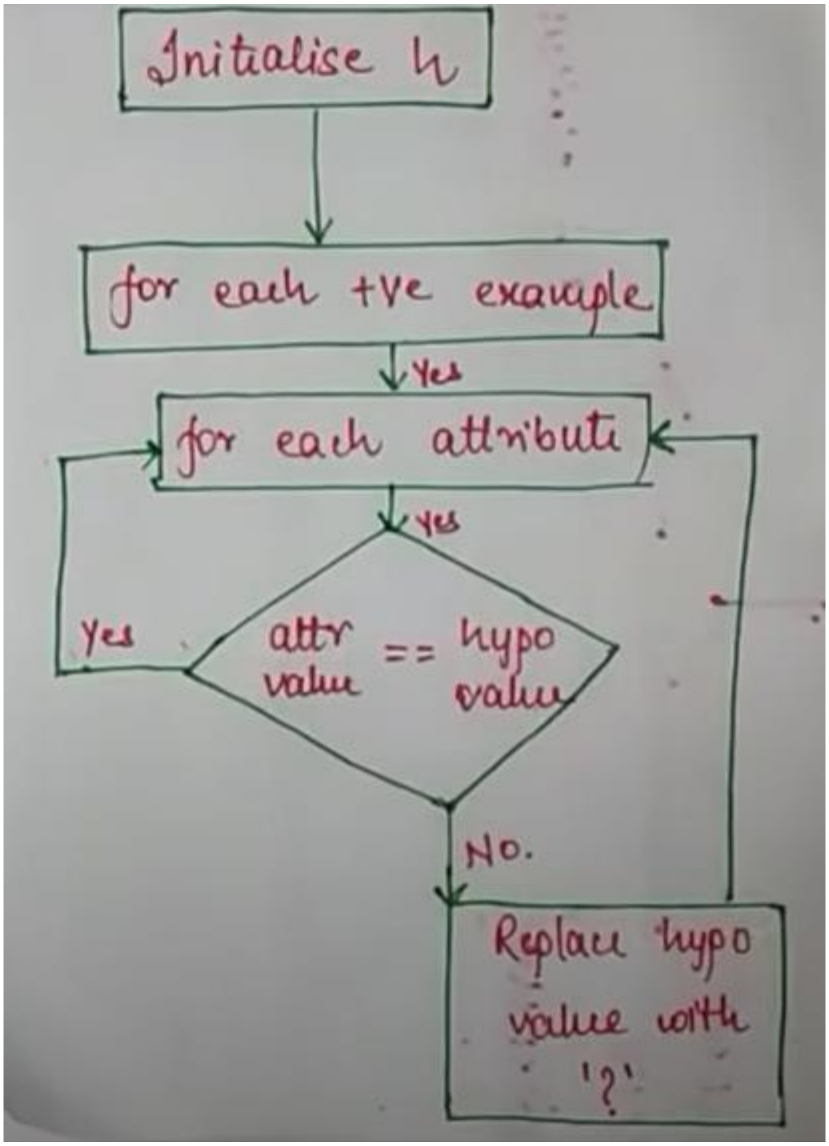
A hypothesis h is **consistent** with a set of training examples D of target concept c if and only if $h(x) = c(x)$ for each training example $\langle x, c(x) \rangle$ in D.

$$Consistent(h, D) \equiv (\forall \langle x, c(x) \rangle \in D)\ h(x) = c(x)$$

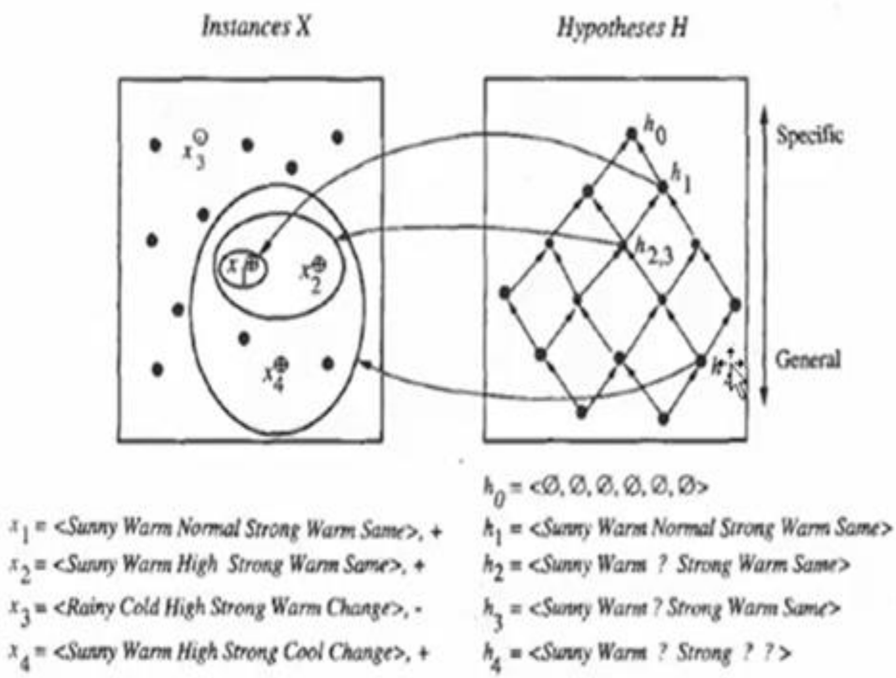
In this algorithm, we have one consistent hypothesis.

The **version space**, $VS_{H,D}$, with respect to hypothesis space H and training examples D, is the subset of hypotheses from H consistent with all training examples in D.

$$VS_{H,D} \equiv \{h \in H | Consistent(h, D)\}$$


Initialise h
for each +ve example
Yes
for each attribute
Yes
Yes
attr value == hypo value
No.
Replace hypo value with '?'

Space Search



The hypothesis space search performed by FIND-S. The search begins (h_0) with the most specific hypothesis in H, then considers increasingly general hypotheses (h_1 through h_4) as mandated by the training examples. In the instance space diagram, positive training examples are denoted by "+," negative by "−," and instances that have not been presented as training examples are denoted by a solid circle.

Limitations:

- Negative samples are not considered
- it outputs just one hypothesis consistent with the training data – there might be many.

To overcome this, we are going to candidate elimination algorithm.

Limitations in the Find S algorithm:

Candidate Elimination Algorithm

To find the consistent hypothesis for the given set of training samples or training examples.

A Concept Learning Task – Enjoy Sport
Training 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	Warm	Change	YES

$\underbrace{\text{Sky, AirTemp, Humidity, Wind, Water, Forecast}}_{\text{ATTRIBUTES}}$ — $\uparrow$ **CONCEPT**

- Consider both positive and negative training samples.
- We will get more than one hypothesis.
- For positive example: tend to generalize specific hypothesis.
- For Negative example: tend to make general hypothesis more specific.

Given Dataset:

Data set

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

Six attributes and target variable is :EnjoySport

Based on the dataset, sometimes person enjoys the sport and sometime he will not enjoy the sport.

person enjoying the sport – positive classification (yes)

if he is not enjoying the sport- negative classification (no)

Algorithm:

- Initialize G & S as most General and specific hypothesis.
- For each example e:

  if e is +ve:
      Make specific hypothesis more general.
  else:
      Make a general hypothesis more specific.

Step1:

The boundary sets are first initialized to G_0 and S_0, the most general and most specific hypotheses in H.

S_0 $\boxed{\langle \varnothing, \varnothing, \varnothing, \varnothing, \varnothing, \varnothing \rangle}$

G_0 $\boxed{\langle ?, ?, ?, ?, ?, ? \rangle}$

Step2:

For the first training sample (x1)

If the training sample is positive(+), steps are similar to FIND- S algorithm.

If +, Make specific hypothesis more general.(Start from start to bottom)

For training example d,

⟨Sunny, Warm, Normal, Strong, Warm, Same ⟩ +

S_0 $\boxed{\langle \varnothing, \varnothing, \varnothing, \varnothing, \varnothing . \varnothing \rangle}$

S_1 $\boxed{\langle \textit{Sunny, Warm, Normal, Strong, Warm, Same} \rangle}$

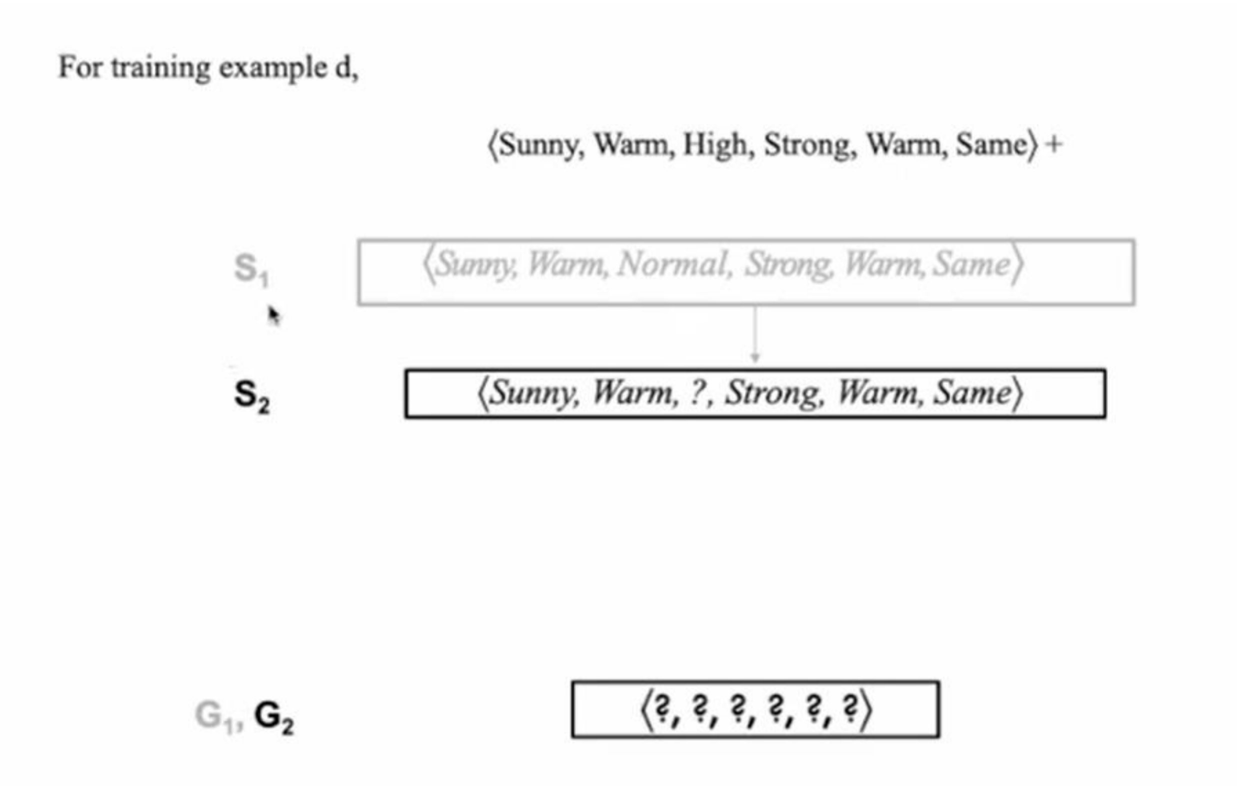
G_0, G_1 $\boxed{\langle ?, ?, ?, ?, ?, ? \rangle}$

We have to check most specific boundary constraints with training sample, if it is inconsistent, replace with next general value. If it is consistent, no change in the hypothesis.

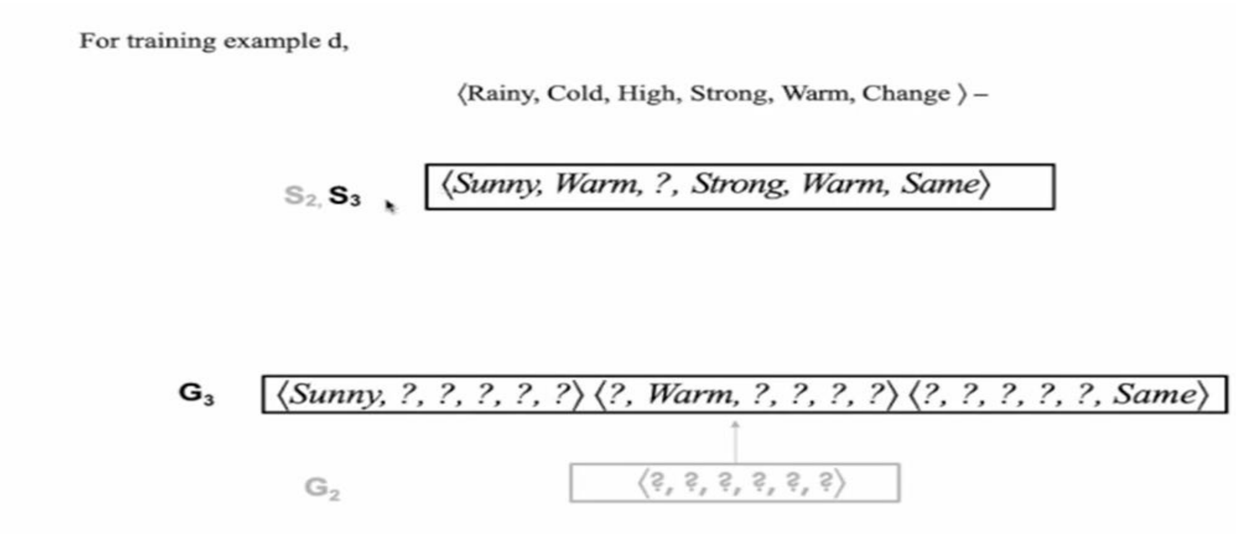
Since it is positive sample (+), no change in the most general boundary. G1 also same.

Take the next training sample.

Step3:

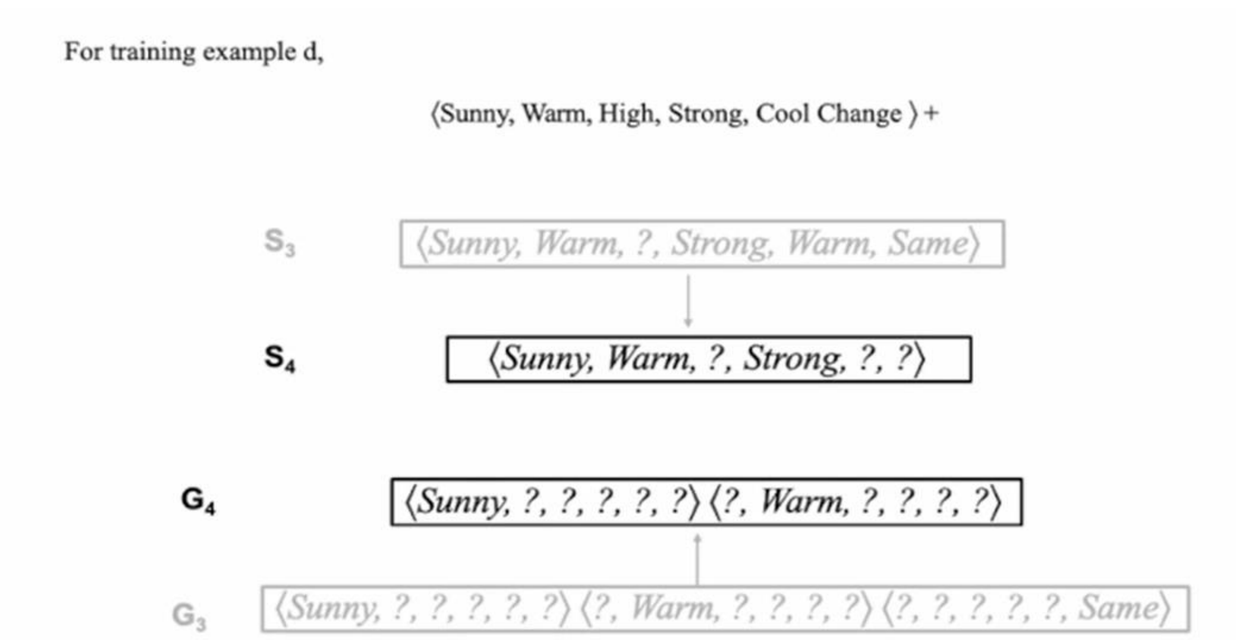


Step4:



To form G3, Compare the training sample d with S3. If any mismatch in both the attribute, write that attribute in G3, remaining all the attributes are '?' as it in G2.

Step5:



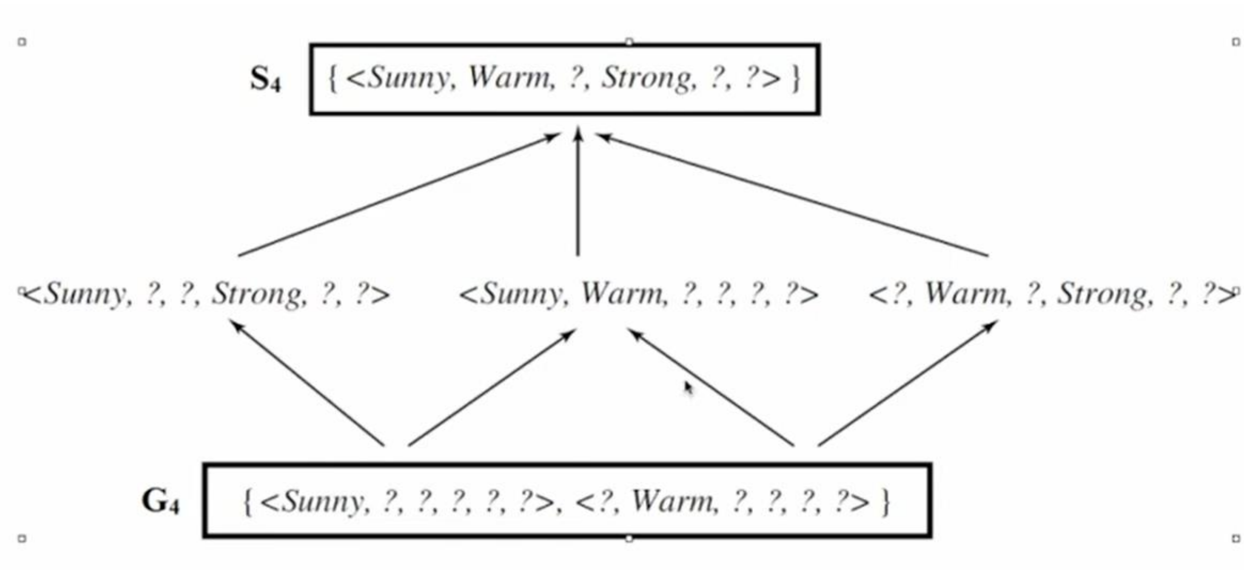
After forming S4, form the G4 also by comparing S4 with G3.

To form G4-> G3 attribute value must be greater than or equal to S4(Because G3 is most general than S4).

In G3 , first two are matching with S4. So we can keep as it is . The third one <?????,same> is not matching with S4. [Because "same" in G3 is lesser than "?" in S4. So we have to remove this]

S4 and G4 are final boundary values. We have to find out hypothesis between of S4 and G4. These hypothesis are called version space.

Version Space:



G4 is compared with S4. If any mismatch, replace with specific value in G4 hypothesis.[Check with every attribute]