Find S Algorithm – It find most specific hypothesis that fits all positive examples. It only considers the positive examples.

This algorithm starts with the most specific hypothesis and generalizes this hypothesis each time it fails to classify an observed positive training data. Hence, the Find-S algorithm moves from the most specific hypothesis to the most general hypothesis.

most specific hypothesis – ϕ (Acceptable)

most general hypothesis – ? (Not Acceptable)

Algorithm :-

Step 1: Initialise with most specific hypothesis (ϕ)

$h_0 = <\phi, \phi, \phi, \phi, \phi> \Rightarrow 5$ attributes

Step 2: For each +ve sample

For each attribute

if (value = hypothesis value)

ignore

else

Replace with most general hypothesis (?)

Numerical Example :-

Origin	*Manufacturer*	*Color*	*Decade*	*Type*	*Example Type*
Japan	Honda	Blue	1980	Economy	Positive
Japan	Toyota	Green	1970	Sports	Negative
Japan	Toyota	Blue	1990	Economy	Positive
USA	Chrysler	Red	1980	Economy	Negative
Japan	Honda	White	1980	Economy	Positive
Japan	Toyota	Green	1980	Economy	Positive
Japan	Honda	Red	1980	Economy	Negative

$h_0 = <\phi, \phi, \phi, \phi, \phi>$

$h_1 = <$ Japan, Honda, Blue, 1980, Economy $>$

$h_2 = h_1$

$h_3 = <$ Japan, ?, Blue, ?, Economy $>$

$h_4 = h_3$

$h_5 = <$ Japan, ?, ?, ?, Economy $>$

$h_6 = <$ Japan, ?, ?, ?, Economy $>$

$h_7 = h_6$

Example 2 :-

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

$h_0 = < \phi, \phi, \phi, \phi, \phi, \phi >$

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

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

$h_3 = h_2$

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

Disadvantages of Find-S :-

- Considers only +ve examples
- It may not be sole hypothesis that fits the complete data.

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

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

For example :-

Example	Citations	Size	InLibrary	Price	Editions	Buy
1	Some	Small	No	Affordable	One	No
2	Many	Big	No	Expensive	Many	Yes

$h_1 = (?, ?, No, ?\ Many)$

For ex. 1, it is not true, and the result is also not true, so it is consistent with ex 1.

For ex. 2, it is true and result is also same.

So, we can say that h_1 is consistent with all examples, so h_1 is consistent.

$h_2 = (?, ?, No, ?, ?)$

For ex. 1, h_2 is consistent, but the result is No, so we can directly say that

h_2 is not consistent as first example itself stands not true.

Version Space :- The Version Space $VS_{H,D}$ is the subset of the hypothesis from H consistent with the training examples in D.

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

H = hypothesis
D = training examples

List - Then - Eliminate Algorithm :- This algorithm is used to obtain the version space.

Step 1: Initially, we will consider all the available hypotheses.

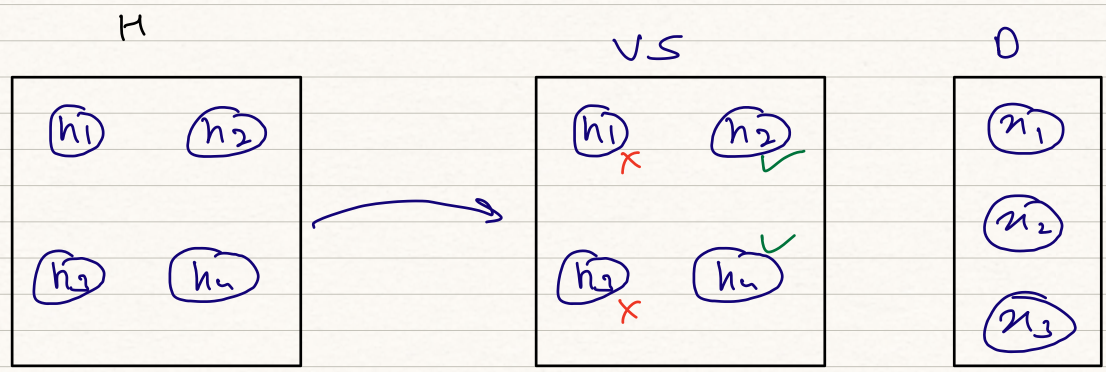
version space ← List containing every hypothesis in H.

Step 2: From this step, we keep on removing inconsistent hypothesis from version space.

For each training example $\langle x, c(x) \rangle$, remove any hypothesis that is $h(x) \neq c(x)$

Step 3: Obtain the list of hypotheses into version space after checking for all the training examples.

Example :-



Now, we will check all the training examples (D).

For example, for h_1,

$$h_1(x_1) = c(x_1) \quad ✓$$

$$h_1(x_2) = c(x_2) \quad ✓$$

$$h_1(x_3) \neq c(x_3) \quad ✗$$

We can see, x_3 is not consistent, so we will eliminate h_1. Same procedure will be done for all the hypothesis.

Example:

F1 – > A, B

F2 – > X, Y

Here F1 and F2 are two features (attributes) with two possible values for each feature or attribute.

Instance Space: (A, X), (A, Y), (B, X), (B, Y) – **4 Examples**

Hypothesis Space: (A, X), (A, Y), (A, **ø**), (A, ?), (B, X), (B, Y), (B, **ø**), (B, ?), (**ø**, X), (**ø**, Y), (**ø**, **ø**), (**ø**, ?), (**?**, X), (**?**, Y), (**?**, **ø**), (**?**, ?) – **16 Hypothesis**

Semantically Distinct Hypothesis : (A, X), (A, Y), (A, ?), (B, X), (B, Y), (B, ?), (**?**, X), (**?**, Y (**?**, ?), (**ø**, **ø**) – **10**

List-Then-Eliminate Algorithm Steps

Version Space: (A, X), (A, Y), (A, ?), (B, X), (B, Y), (B, ?), (?, X), (?, Y) (?, ?), (**ø**, **ø**), •Training Instances

F1	F2	Target
A	X	Yes
A	Y	Yes

Here, consistent hypothesis are (version space): (A, ?), (?, ?)

Candidate Elimination Algorithm:-

- Uses the concept of Version Space
- Considers both +ve and -ve values (yes and no).
- We will get both specific and general hypothesis.
- It is the extended version of Find-S Algorithm.

For positive samples, move from specific to general.

For negative samples, move from general to specific.

$S = \{ \phi, \phi, \phi, \phi, \phi \}$ + ↓

$G = \{ ?, ?, ?, ?, ? \}$ − ↑

Algorithm:-

Step 1: Load the dataset.

Step 2: Initialize general and specific hypothesis.

$S = \{ \phi, \phi, \phi, \dots \phi \}$ Depends on no. of attributes.
$G = \{ ?, ?, ?, \dots, ? \}$

Step 3: If example is positive (+ve):

if attribute_value == hypothesis value
do nothing
else:
replace attribute_value with '?'
(basically generalizing it)

else:
Make generalized hypothesis more specific.

Example:-

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	Rain	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

$S_0 = \{ \phi, \phi, \phi, \phi, \phi, \phi \}, \quad G_0 = \{ ?, ?, ?, ?, ?, ? \}$

1) +ve,

$S_1 = \{$ Sunny, Warm, Normal, Strong, Warm, Same $\}$
$G_1 = \{ ?, ?, ?, ?, ?, ? \}$

2) +ve, (Specific to General)

$S_2 = \{$ Sunny, Warm, ?, Strong, Warm, Same $\}$
$G_2 = \{ ?, ?, ?, ?, ?, ? \}$

3) −ve (keep specific hypothesis as it is.)

S_3 = {Sunny, warm, ? , Strong, Warm, Same}

G_3 = {<'Sunny', ?, ?, ?, ?, ?>, <?, 'Warm', ?, ?, ?, ?>,
<?, ?, ?, ?, ?, 'same'>}

4) +ve

S_4 = {'Sunny', 'Warm', ?, 'Strong', ?, ?}

G_4 = {<'Sunny', ?, ?, ?, ?, ?>, <?, 'Warm', ?, ?, ?, ?>}

Here, if you notice, we have removed third attribute (same) from G_4, because it is not present in Specific hypothesis.

Inductive Learning :- Here, from examples we devieve rules. It is also known as Concept Learning, it is how AI systems attempts to use a generalized rules to carry out observations.

Deductive Learning :- Here, existing rules are applied to our examples

Biased Hypothesis Space :- does not consider all types of training example.

Solution − Include all hypothesis.

Unbiased Hypothesis Space :- Providing a hypothesis capable of representing set of all examples.

Inductive Bias :- Based on some previous examples, machine will find a generalized rule, and it will apply this rule to all the new examples

Learner generalizes beyond the observed training examples to infer new examples.

'>' → Inductively infered from

$x > y \Rightarrow y$ is inductively infered from x.

Inductive bias is the set of assumptions a learner uses to predict results on given inputs it has not yet encountered. It has some prior assumptions about the task.

An inductive bias allows a learning algorithm to prioritize one solution (or interpretation) over another.