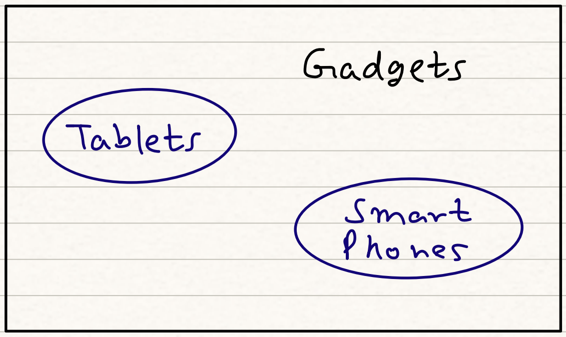
Concept Learning - Concepts in Machine Learning can be thought of as a boolean-valued function defined over a set of training data.

Concept Learning can be formulated as a problem of searching through a predefined space of potential hypothesis for the hypothesis that best fits the training examples.

It is a task of acquiring potential hypothesis (solution) that best fits the given training examples.



Universe

* Features (binary valued attributes)

Size : Large, Small → x1

Color : Black, Blue → x2

ScreenType : Flat, Folded → x3

Shape : Square, Rectangle → x4

Concept = < x1, x2, x3, x4 >

Tablet : < Large, Black, Flat, Square >

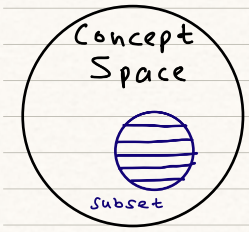
Smartphone : < Small, Blue, Folded, Rectangle >

No. of possible instances : 2^d $2^4 = 16$

(where d = no. of features)

Total Possible Concepts : 2^{2^d} 2^{16}

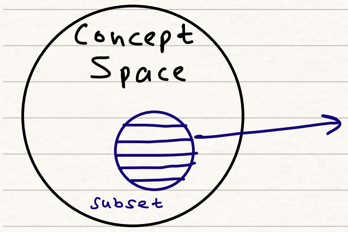
We are not going to learn about 2^{16} concepts. We will only choose the consistent concepts.



From large concept space, we are only considering subset.

Reducing a concept space to a smaller region for study is called as Inductive Bias.

The inductive bias of a learning algorithm is the set of assumptions that the learner uses to predict outputs of a given input that it has not encountered.



Target Concept / Hypothesis Space.

Most Specific Hypothesis $< \phi , \phi , \phi , \phi >$ → Reject All

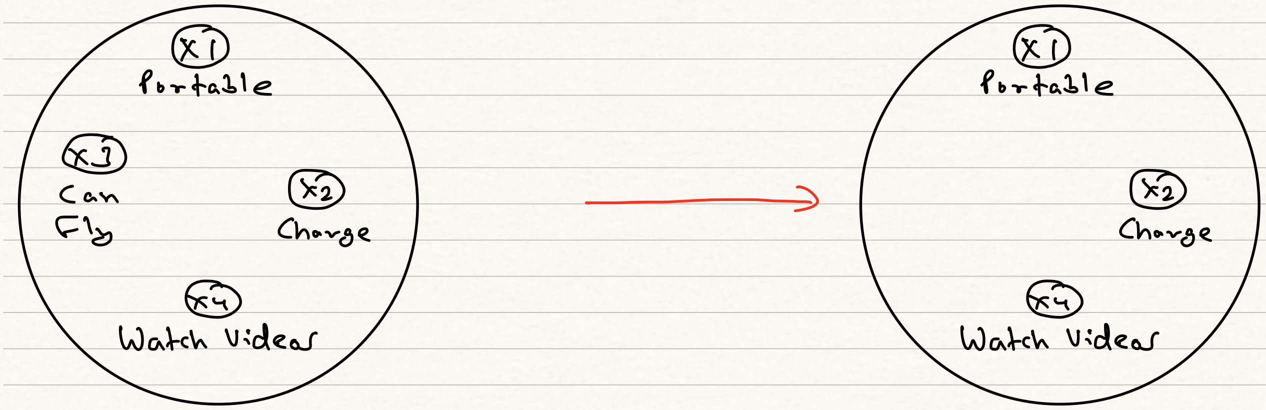
Most General Hypothesis $< ? , ? , ? , ? >$ → Accept All

Goal of Concept Learning - Find all concepts / hypothesis that are consistent.

$$h(x) = c(x)$$

Mobile Phones

1990



Hypothesis

hypothesis is a function that best describes the target in supervised ML.

Version Space

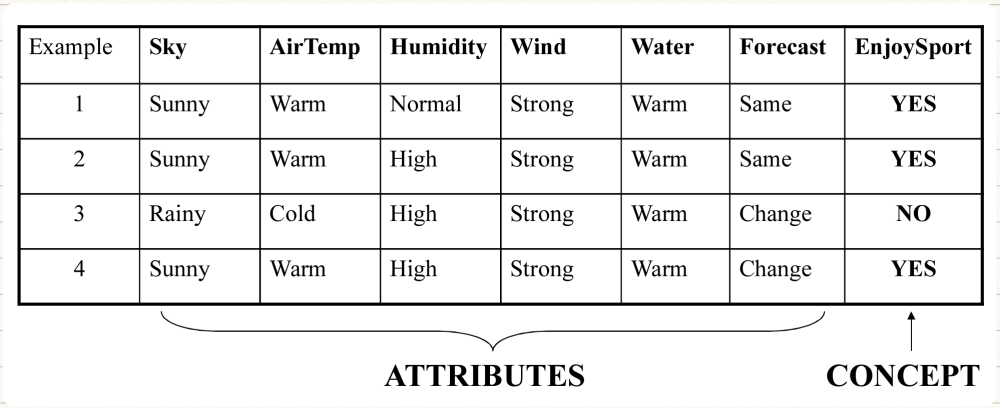
(A predefined space of hypothesis).

In Concept Learning task, a human or machine learner is trained to classify objects by being shown a set of example objects along with their class labels.

Active Learning - A special case of ML where a learning algorithm can interactively query a user to label new data points with the desired output.

AL - Reduction in Search Space
ML - Predection, Accuracy

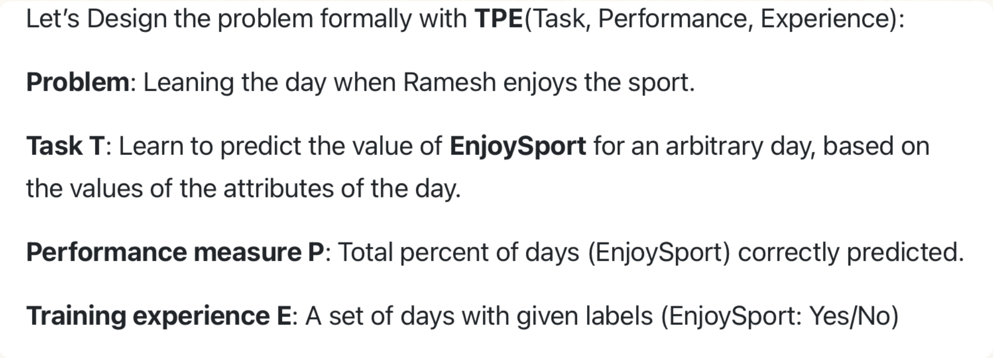
Concept Learning Task - One example of possible target concept may be to find the day when my friend John enjoy his favourite sports.



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**

ATTRIBUTES

CONCEPT



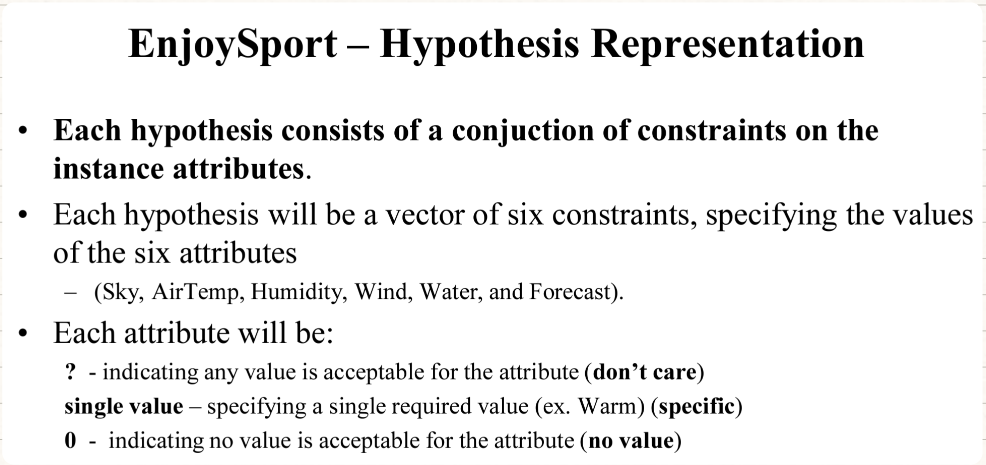
Let's Design the problem formally with **TPE**(Task, Performance, Experience):

Problem: Leaning the day when Ramesh enjoys the sport.

Task T: Learn to predict the value of **EnjoySport** for an arbitrary day, based on the values of the attributes of the day.

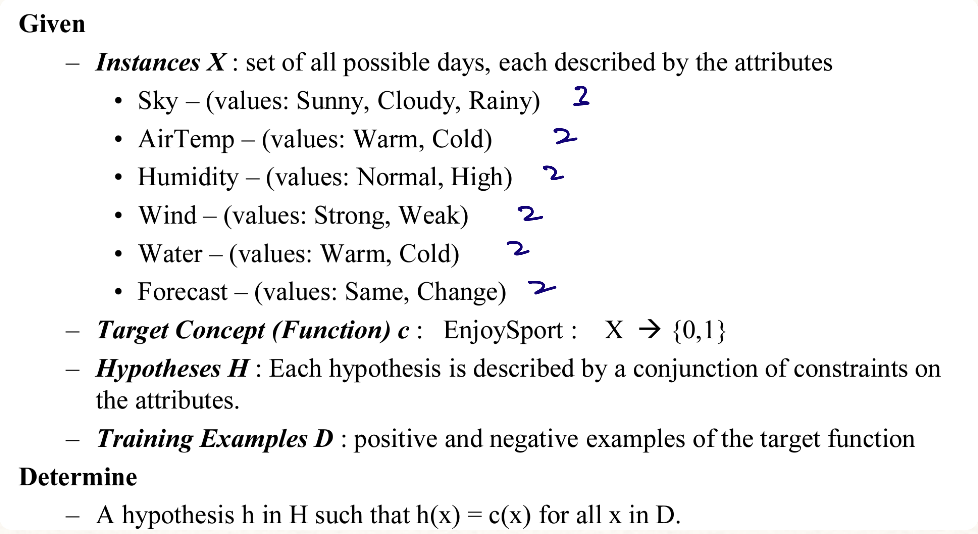
Performance measure P: Total percent of days (EnjoySport) correctly predicted.

Training experience E: A set of days with given labels (EnjoySport: Yes/No)



EnjoySport – Hypothesis Representation

- **Each hypothesis consists of a conjuction of constraints on the instance attributes**.
- Each hypothesis will be a vector of six constraints, specifying the values of the six attributes
  - (Sky, AirTemp, Humidity, Wind, Water, and Forecast).
- Each attribute will be:
  - **?** - indicating any value is acceptable for the attribute (**don't care**)
  - **single value** – specifying a single required value (ex. Warm) (**specific**)
  - **0** - indicating no value is acceptable for the attribute (**no value**)

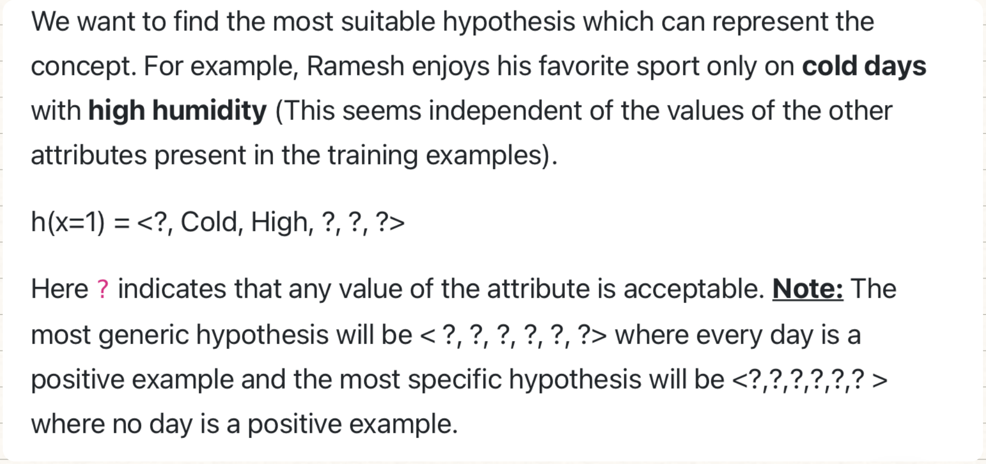


Given

- *Instances X* : set of all possible days, each described by the attributes
  - Sky – (values: Sunny, Cloudy, Rainy) 2
  - AirTemp – (values: Warm, Cold) 2
  - Humidity – (values: Normal, High) 2
  - Wind – (values: Strong, Weak) 2
  - Water – (values: Warm, Cold) 2
  - Forecast – (values: Same, Change) 2
- *Target Concept (Function) c* : EnjoySport : X → {0,1}
- *Hypotheses H* : Each hypothesis is described by a conjunction of constraints on the attributes.
- *Training Examples D* : positive and negative examples of the target function

Determine

- A hypothesis h in H such that h(x) = c(x) for all x in D.



We want to find the most suitable hypothesis which can represent the concept. For example, Ramesh enjoys his favorite sport only on **cold days** with **high humidity** (This seems independent of the values of the other attributes present in the training examples).

h(x=1) = <?, Cold, High, ?, ?, ?>

Here ? indicates that any value of the attribute is acceptable. **Note:** The most generic hypothesis will be < ?, ?, ?, ?, ?, ?> where every day is a positive example and the most specific hypothesis will be <?,?,?,?,?,? > where no day is a positive example.

Concept Learning as Search :- Concept Learning can be viewed as the task of searching through a large space of hypotheses and find the hypothesis that best fits the training example.

In the EnjoySports example, we have six attributes:

(Sky, Air Temperature, Humidity, Wind, Water, Forecast)

here, only sky has 3 values i.e. Rainy, Cloudy & Sunny

, remaining attributes have only two values.

So, total instances possible : 3 x 2 x 2 x 2 x 2 x 2 = 96

Syntactically Distinct Hypothesis → For each and every attribute, we aditionally add ? and ϕ.

? ↓ Accept All (Most General)

ϕ → Reject All (Most Specific)

Example of Sky attribute in syntactically distinct hypothesis will be :-

< ϕ , Rainy, Cloudy, Sunny, ? >

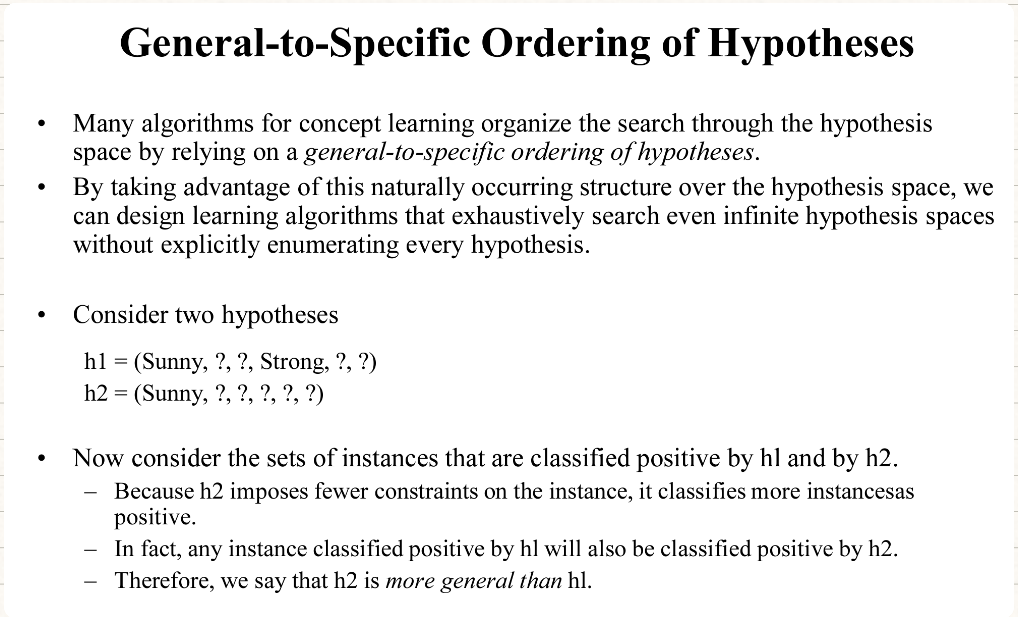
Same process will be done for all other attributes.

So, total syntactically distinct hypothesis = $5 \times 4 \times 4 \times 4 \times 4 \times 4$
$= 5120$

Semantically Distinct Hypothesis → Here, instead of adding ϕ to each and every attribute, we will take it as common.

Result will be - $1 + (4 \times 3 \times 3 \times 3 \times 3 \times 3) = 973$

After finding all the syntactically and semantically distinct hypothesis, we search for the best match from all these (i.e. much closer to our learning problem).



General-to-Specific Ordering of Hypotheses

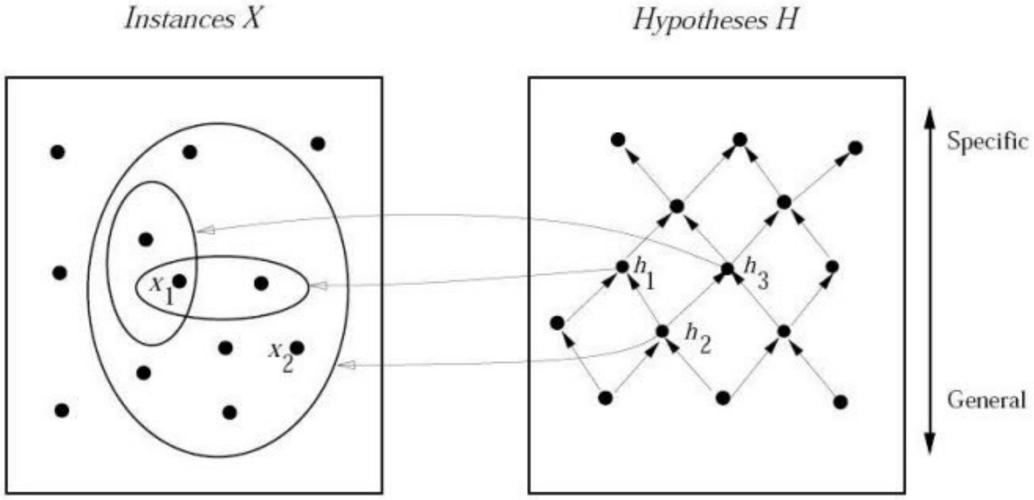
- Many algorithms for concept learning organize the search through the hypothesis space by relying on a *general-to-specific ordering of hypotheses*.
- By taking advantage of this naturally occurring structure over the hypothesis space, we can design learning algorithms that exhaustively search even infinite hypothesis spaces without explicitly enumerating every hypothesis.
- Consider two hypotheses

  h1 = (Sunny, ?, ?, Strong, ?, ?)
  h2 = (Sunny, ?, ?, ?, ?, ?)

- Now consider the sets of instances that are classified positive by hl and by h2.
  - Because h2 imposes fewer constraints on the instance, it classifies more instancesas positive.
  - In fact, any instance classified positive by hl will also be classified positive by h2.
  - Therefore, we say that h2 is *more general than* hl.

More-General-Than Relation

- For any instance x in X and hypothesis h in H, we say that x satisfies h if and only if h(x) = 1.
- **More-General-Than-Or-Equal Relation:**
  Let h1 and h2 be two boolean-valued functions defined over X.
  Then h1 is ***more-general-than-or-equal-to*** h2 (written h1 ≥ h2)
  if and only if any instance that satisfies h2 also satisfies h1.
- h1 is ***more-general-than*** h2 (h1 > h2) if and only if h1≥h2 is true and h2≥h1 is false. We also say h2 is ***more-specific-than*** h1.



x_1= <Sunny, Warm, High, Strong, Cool, Same>
x_2= <Sunny, Warm, High, Light, Warm, Same>

h_1= <Sunny, ?, ?, Strong, ?, ?>
h_2= <Sunny, ?, ?, ?, ?, ?>
h_3= <Sunny, ?, ?, ?, Cool, ?>

- In the figure, the box on the left represents the set X of all instances, the box on the right the set H of all hypotheses.
- Each hypothesis corresponds to some subset of X, that is, the subset of instances that it classifies positive. In the figure, there are 2 instances x1 and x2, and 3 hypothesis h1, h2 and h2.
- h1 classifies — x1, h2 classifies — x1 and x2, and h3 classifies — x1. This indicates, h2 is more-general-than h1 and h3.
- The arrows connecting hypotheses represent the more-general-than relation, with the arrow pointing toward the less general hypothesis.
- Note the subset of instances characterized by h2 subsumes the subset characterized by hl , hence h2 is more-general–than h1

Till now, we performed, ordering the hypothesis in hypothesis space from specific to general. Many learning algorithms for concept learning organize the search through the hypothesis space by relying on a general-to-specific ordering of hypotheses. By taking advantage of this naturally occurring structure over the hypothesis space, we can design learning algorithms that exhaustively search even infinite hypothesis spaces without explicitly enumerating every hypothesis.