# Administrative Issues

- Syllabus handed out in class today
- Reading for next class: Mitchell, chapter 7.
- No class on Wednesday, January 17.

# cture 2: Learning (Boolean) Concepts and Inductive Bias

- What is concept learning?
- Learning as search
- Version Spaces
- Inductive bias and why we cannot live without it
- Kinds of biases

# Associative learning

Learner is given pairs of objects and is asked to remember their association

Based on this data, the learner will later be asked to predict the second object given the first one, for objects that have not been seen before.

Examples:

- Concept learning: (instance, label)
- System identification (input, output)

# Two kinds of prediction problems

- Single-step: (*classification, function approximation*)
  - All the information necessary to identify the second object is available right away
  - During the training phase, the correct answers are also provided to the learner
- Multi-step: (*reinforcement learning*)
  - A sequence of observations is made
  - The outcome is revealed only at the end

# Concept learning

*Given: a set of examples $E$ labelled as part (or not part) of the concept. and a representation language*

*Infer a description of the concept in the representation language*

The description is a Boolean-valued function (returns `true` or 1 if an object is part of the concept and `false` or 0 otherwise).

Example: the "chair" concept

- The representation language is a set of *features* (or *attributes*) for describing chairs, such as `number-of-legs`, `has-back` etc.
- The training examples are of the form:
  `number-of-legs=4` $\wedge$ `has-back` $\wedge$ `...  1`
- The description of the concept is a conjunction of tests on the Boolean features

# A simple example: *EnjoySport*

| 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 |

No noise, perfect data.

What is the general concept?

# Representing Hypotheses

Many possible representations!

Here, $h$ is conjunction of constraints on attributes

Each constraint can be

- a specfic value (e.g., $Water = Warm$)
- don't care (e.g., "$Water =?$")
- no value allowed (e.g.,"Water=$\emptyset$")

For example,

| Sky | AirTemp | Humid | Wind | Water | Forecst |
|---|---|---|---|---|---|
| $\langle Sunny$ | $?$ | $?$ | $Strong$ | $?$ | $Same\rangle$ |

# Prototypical Concept Learning Task

- **Given:**
  - Instances $X$: Possible days, each described by the attributes *Sky*, *AirTemp*, *Humidity*, *Wind*, *Water*, *Forecast*
  - Target function $c$: $EnjoySport : X \rightarrow \{0, 1\}$
  - Hypotheses $H$: Conjunctions of literals. E.g.

    $$\langle ?, Cold, High, ?, ?, ? \rangle.$$

  - Training examples $D$: Positive and negative examples of the target function

    $$\langle x_1, c(x_1) \rangle, \ldots \langle x_m, c(x_m) \rangle$$

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

# 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 examples.*

Inductive learning assumes that the examples are drawn from some set $X$ using an unknown *but fixed* probability distribution $P$

*Unless the unseen examples are drawn from* $P$*, no guarantees can be made about any inductive learning algorithm!*

# The simplest approach: learning by enumeration

1. Construct a list of all possible hypotheses
2. Eliminate all the hypothesis that are not consistent with all the training examples
3. Select one of the remaining hypotheses as the description

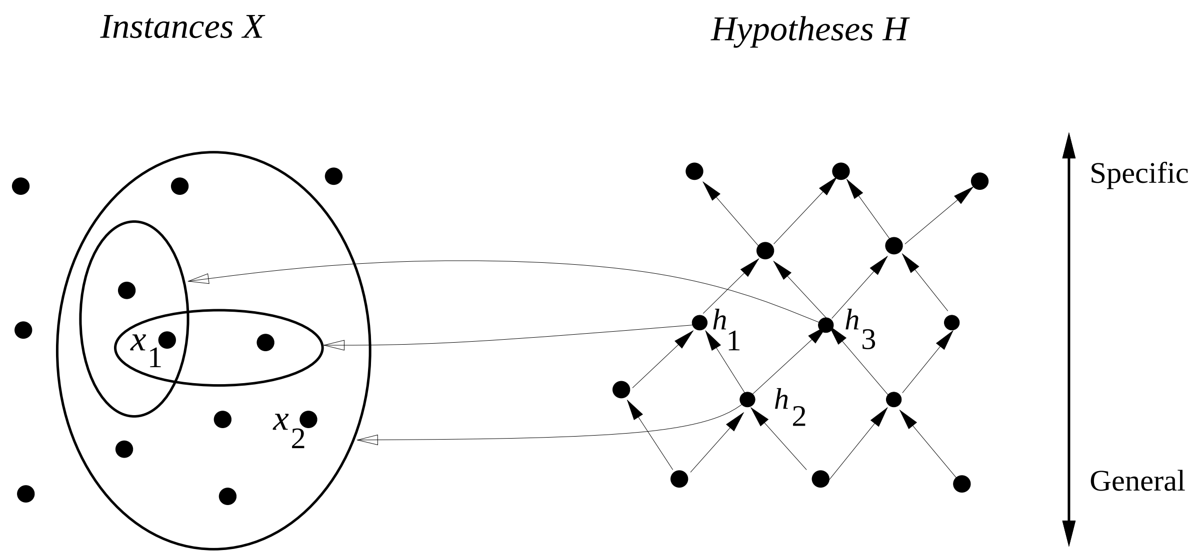
*What if the list of hypotheses is very large or infinite?*

**Search through the hypotheses space!**

# Learning as search

- The space of hypotheses is defined by the representation language
- A learning algorithm can be viewed as searching this space for a hypotheses consistent with the training data
- The size of the hypothesis space determines the difficulty of the learning problem
- The more expressive the language, the more problems we can solve, but the more search effort needs to be expanded
- Like in any AI problem, use heuristicsa to guide the search

# General-to-specific ordering of hypotheses




$x_1$ = <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, ?>

A hypotheses $h$ is more general than or equal to another hypothesis $h'$ iff

$$\forall x \in X h'(x) \rightarrow h(x)$$

# Version Spaces

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 $D$.

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

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 $D$.

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

**Key idea:** version spaces can be represented by keeping track ofthe maximally general and maximally specific members of the version space

# Representing Version Spaces

The **General boundary**, G, of version space $VS_{H,D}$ is the set of its maximally general members

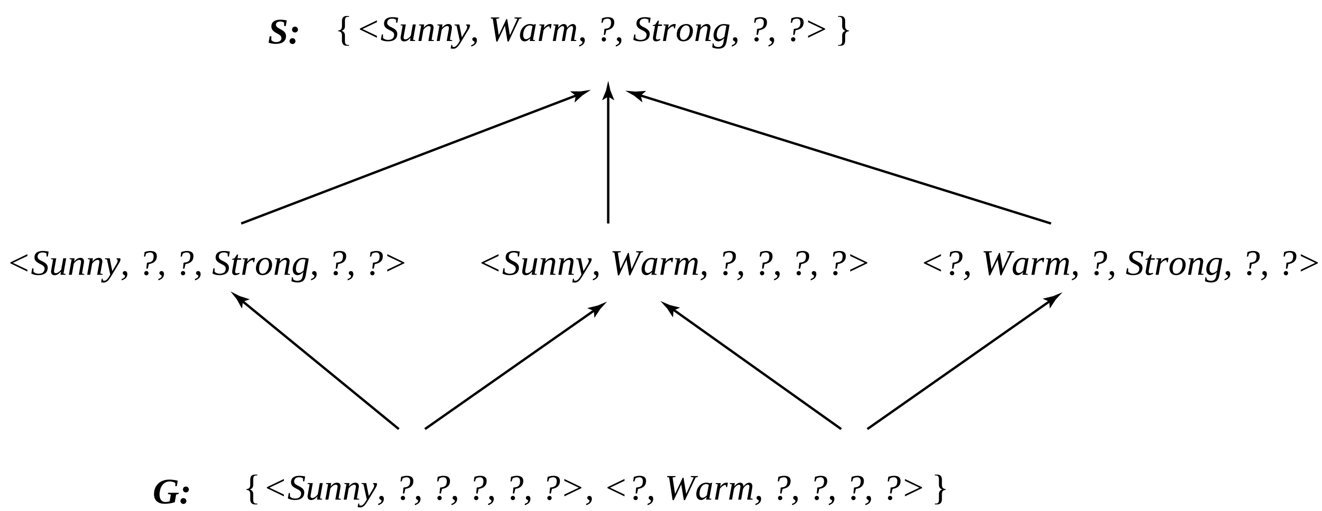
The **Specific boundary**, S, of version space $VS_{H,D}$ is the set of its maximally specific members

Every member of the version space lies between these boundaries

$$VS_{H,D} = \{h \in H | (\exists s \in S)(\exists g \in G)(g \geq h \geq s)\}$$

where $x \geq y$ means $x$ is more general or equal to $y$

# Example Version Space



# Candidate Elimination Algorithm

$G$ maximally general hypotheses in $H$

$S$ maximally specific hypotheses in $H$

For each training example $d$, do

1. If $d$ is a positive example
   (a) Remove from $G$ any hypothesis inconsistent with $d$
   (b) For each hypothesis $s$ in $S$ that is not consistent with $d$
      i. Remove $s$ from $S$
      ii. Add to $S$ all minimal generalizations $h$ of $s$ such that
         A. $h$ is consistent with $d$, and
         B. some member of $G$ is more general than $h$
      iii. Remove from $S$ any hypothesis that is more general than another hypothesis in $S$
2. If $d$ is a negative example
   (a) Remove from $S$ any hypothesis inconsistent with $d$

(b) For each hypothesis $g$ in $G$ that is not consistent with $d$

  i. Remove $g$ from $G$

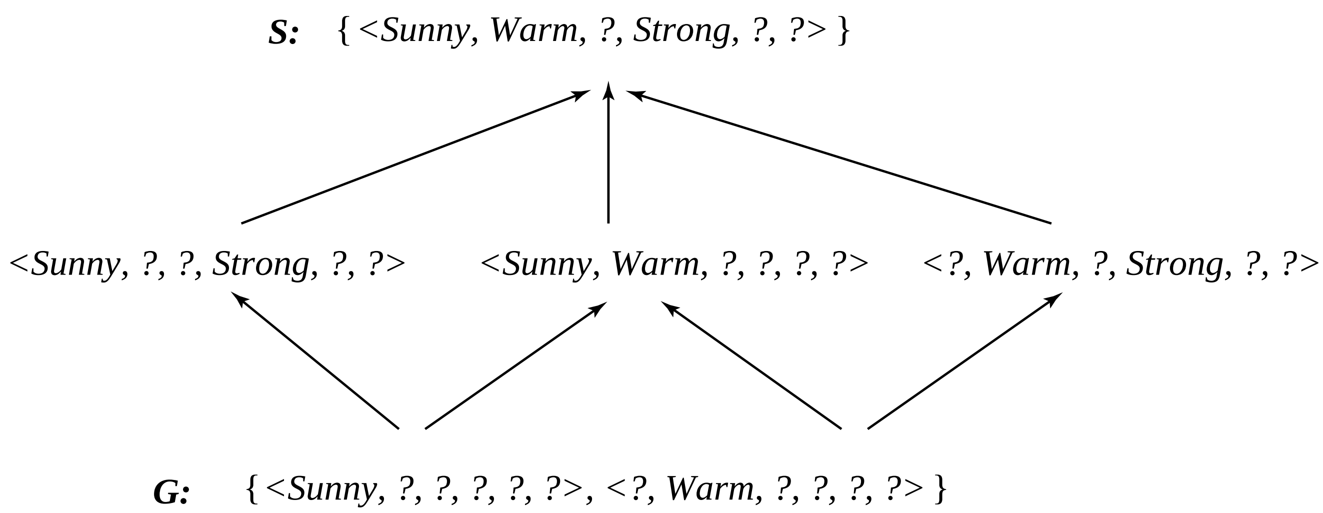
  ii. Add to $G$ all minimal specializations $h$ of $g$ such that

    A. $h$ is consistent with $d$, and

    B. some member of $S$ is more specific than $h$

  iii. Remove from $G$ any hypothesis that is less general than another hypothesis in $G$

# What Next Training Example?




An instance that would be classified positively by some hypotheses and negatively by others

# How Should These Be Classified?

$\langle Sunny\ Warm\ Normal\ Strong\ Cool\ Change \rangle$

$\langle Rainy\ Cool\ Normal\ Light\ Warm\ Same \rangle$

$\langle Sunny\ Warm\ Normal\ Light\ Warm\ Same \rangle$

# The good and bad of version spaces

Good news

- Guaranteed to converge to the correct target concept assuming that the data is error-free and that the target concept is in the hypotheses space
- *Incremental!* Allows continual uupdating as examples become available, and also allws choosing the next training example
- Can classify unseen examples based on partially learned concepts
- Can be used with different hypotheses spaces

Bad news

- Can be VERY VERY VERY slow!
- The S and G sets can grow very large, depending on the order in which examples are presented

# An UNBiased Learner

Idea: Choose $H$ that expresses every teachable concept (i.e., $H$ is the power set of $X$)

Consider $H' =$ disjunctions, conjunctions, negations over previous $H$. E.g.,

$$\langle Sunny\ Warm\ Normal\ ?\ ?\ ?\rangle \ \vee\ \neg\langle ?\ ?\ ?\ ?\ ?\ Change\rangle$$

# Inductive Bias

Consider

- concept learning algorithm $L$
- instances $X$, target concept $c$
- training examples $D_c = \{\langle x, c(x) \rangle\}$
- let $L(x_i, D_c)$ denote the classification assigned to the instance $x_i$ by $L$ after training on data $D_c$.

**Definition**:

The **inductive bias** of $L$ is any minimal set of assertions $B$ such that for any target concept $c$ and corresponding training examples $D_c$

$$(\forall x_i \in X)[(B \wedge D_c \wedge x_i) \rightarrow L(x_i, D_c)]$$

where $A \rightarrow B$ means $A$ logically entails $B$

# Types of inductive bias

- Absolute bias
  - Rules out part of the hypotheses space and picks hypotheses from the rest
  - Defines awaht can be learned (the hypotheses space)
  - E.g. choose a restricted language
  - E.g. assign prior probabilities on hypotheses
- Preference Bias
  - Controls how the search proceeds
  - Choose the simplest hypotheses (Occam's rasor)
  - E.g. Ordering the hypotheses by syntactic simplicity