Specifying A Learning Problem

- **Learning = Improving with Experience at Some Task**
  - **Improve over task *T,***
  - **with respect to performance measure *P*,**
  - **based on experience *E*.**

- **Refining the Problem Specification: Issues**
  - **What experience?**
  - **What *exactly* should be learned?**
  - **How shall it be *represented*?**
  - **What specific algorithm to learn it?**

Example (Revisited): Learning to Play Board Games

- **Type of Training Experience**
  - **Direct or indirect?**
  - **Teacher or not?**
  - **How well distributed are the Training Examples?**

$\hat{V}$

<u>Performance Element</u>: What to Learn?

- Classification Functions
  - Hidden functions: estimating ("fitting") parameters
  - <u>Concepts</u> (e.g., chair, face, game)
  - Diagnosis, prognosis: medical, risk assessment, fraud, mechanical systems
- Models
  - Map (for navigation)
  - Distribution (query answering, *aka* QA)
  - Language model (e.g., automaton/grammar)
- Skills
  - Playing games
  - Planning
  - <u>Reasoning</u> (acquiring representation to use in reasoning)
- Cluster Definitions for Pattern Recognition
  - Shapes of objects
  - Functional or taxonomic definition
- *Many Learning Problems Can Be Reduced to Classification*

(Supervised) Concept Learning

- Given: Training Examples <*x, f(x)*> of Some Unknown Function *f*
- Find: A Good Approximation to *f*
- Examples (besides Concept Learning)
  - Disease diagnosis
    - *x* = properties of patient (medical history, symptoms, lab tests)
    - *f* = disease (or recommended therapy)
  - Risk assessment
    - *x* = properties of consumer, policyholder (demographics, accident history)
    - *f* = risk level (expected cost)
  - Automatic steering
    - *x* = bitmap picture of road surface in front of vehicle
    - *f* = degrees to turn the steering wheel
  - Part-of-speech tagging
  - Fraud/intrusion detection
  - Web log analysis
  - Multisensor integration and prediction

Example:
Learning A Concept (*EnjoySport*) from Data

- **Specification for Training Examples**
  - **Similar to a data type definition**
  - **6 variables (*aka* <u>attributes</u>, <u>features</u>):**
    Sky*, *Temp*, *Humidity*, *Wind*, *Water*, *Forecast
  - **<u>Nominal</u>-valued (<u>symbolic</u>) attributes - enumerative data type**
- **Binary (Boolean-Valued or H -Valued) Concept**
- **Supervised Learning Problem: *Describe the General Concept***

Example	Sky	Air Temp	Humidity	Wind	Water	Forecast	Enjoy Sport
0	Sunny	Warm	Normal	Strong	Warm	Same	**Yes**
1	Sunny	Warm	High	Strong	Warm	Same	**Yes**
2	Rainy	Cold	High	Strong	Warm	Change	**No**
3	Sunny	Warm	High	Strong	Cool	Change	**Yes**

Representing Hypotheses

- **Many Possible Representations**
- **Hypothesis *h*: Conjunction of Constraints on Attributes**
- **Constraint Values**
  - **Specific value (e.g., *Water* = *Warm*)**
  - **Don't care (e.g., *"Water = ?"*)**
  - **No value allowed (e.g., "Water = Ø")**
- **Example Hypothesis for *EnjoySport***
Sky	**AirTemp**	**Humidity**	**Wind**	**Water**	**Forecast**
<*Sunny*	***?***	***?***	***Strong***	***?***	***Same*>**
  - **Is this consistent with the training examples?**
  - **What are some hypotheses that are consistent with the examples?**

Typical Concept Learning Tasks

- **Given**
  - **Instances X: possible days, each described by attributes *Sky, AirTemp, Humidity, Wind, Water, Forecast***
  - **Target function *c* ≡ EnjoySport: $X \rightarrow \mathrm{H} \equiv$ {{Rainy, Sunny} × {Warm, Cold} × {Normal, High} × {None, Mild, Strong} × {Cool, Warm} × {Same, Change}} → {0, 1}**
  - **Hypotheses *H*: conjunctions of literals (e.g., <*?, Cold, High, ?, ?, ?*>)**
  - **Training examples *D*: positive and negative examples of the target function**

$$\langle x_{1,} c(x_1) \rangle, \ldots, \langle x_{m,} c(x_m) \rangle$$

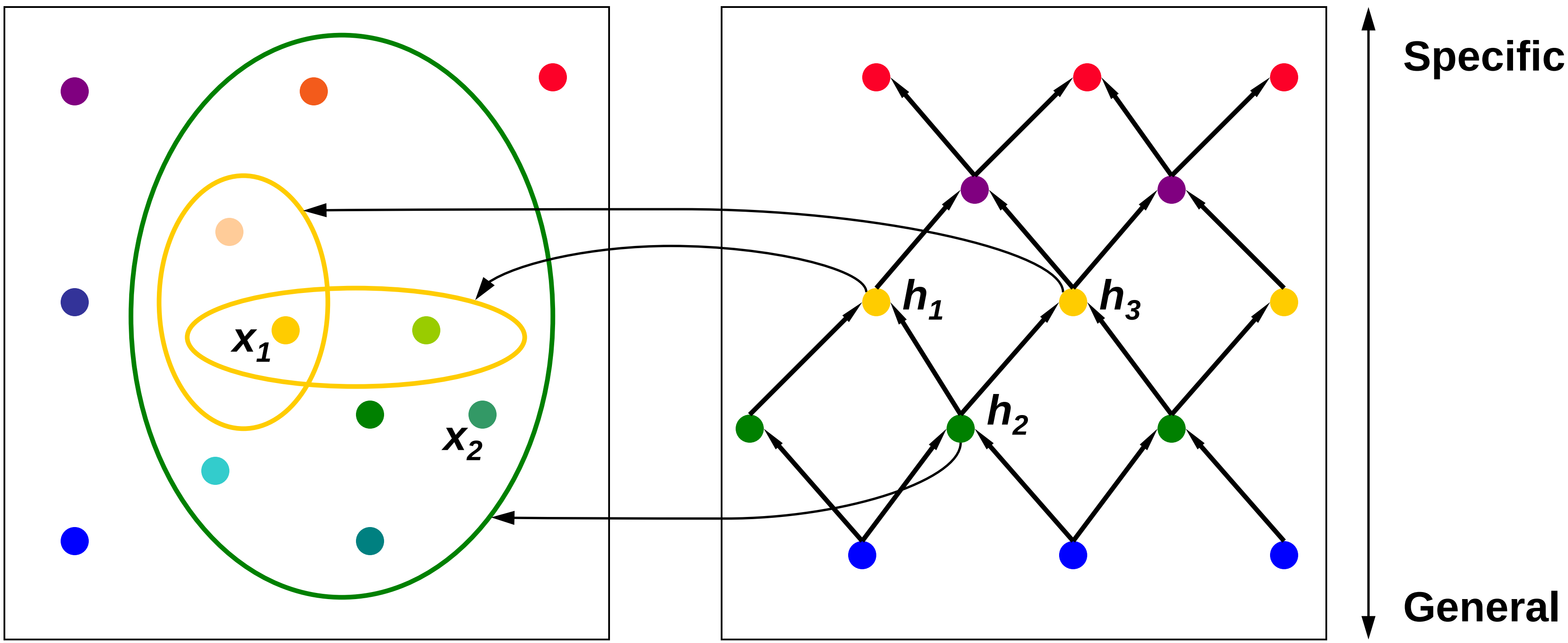
- **Determine**
  - **Hypothesis *h* $\in$ *H* such that *h(x)* = *c(x)* for all *x* $\in$ *D***
  - **Such *h* are consistent with the training data**
- **Training Examples**
  - **Assumption: no missing *X* values**
  - **Noise in values of *c* (contradictory labels)?**

Inductive Learning Hypothesis

- **Fundamental Assumption of Inductive Learning**
- **Informal Statement**
  - **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**
  - **Definitions deferred: *sufficiently large, approximate well, unobserved***
- **Next: How to *Find* This Hypothesis?**

Instances, Hypotheses, and the Partial Ordering *Less-Specific-Than*

Instances *X* **Hypotheses *H***



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, ?>

$\leq_P$ ≡ ***Less-Specific-Than*** ≡ ***More-General-Than***

$h_2 \leq_P h_1$
$h_2 \leq_P h_3$

Find-S Algorithm

1. Initialize h to the most specific hypothesis in H

    H: the *hypothesis space* (partially ordered set under relation *Less-Specific-Than*)

2. For each positive training instance x

    For each attribute constraint a_i in h

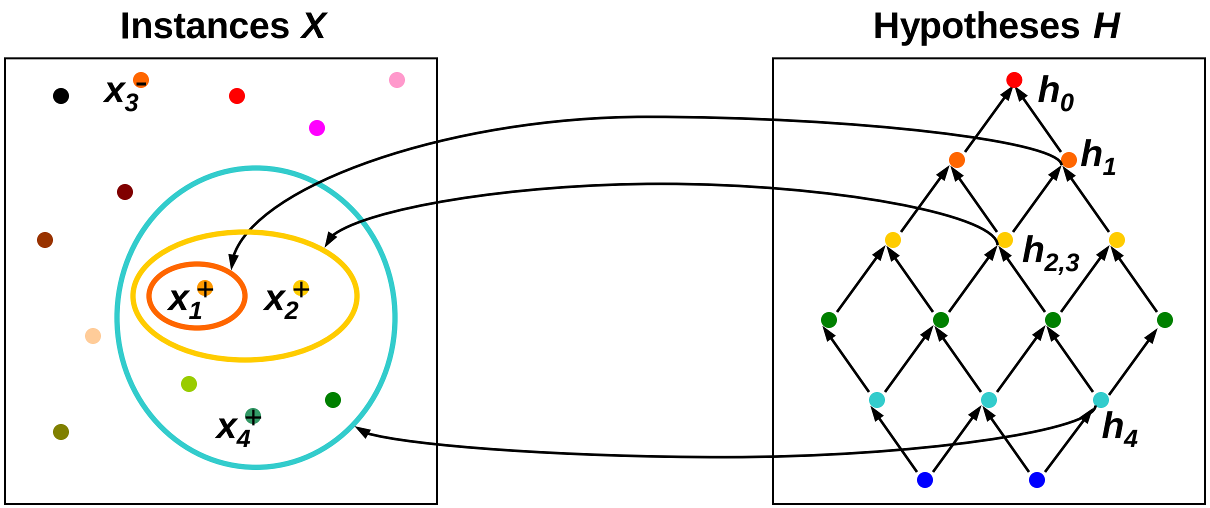
        IF the constraint a_i in h is satisfied by x

        THEN do nothing

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

3. Output hypothesis h

Hypothesis Space Search by *Find-S*



x_1 = <Sunny, Warm, Normal, Strong, Warm, Same>, +
x_2 = <Sunny, Warm, High, Strong, Warm, Same>, +
x_3 = <Rainy, Cold, High, Strong, Warm, Change>, -
x_4 = <Sunny, Warm, High, Strong, Cool, Change>, +

h_1 = <Ø, Ø, Ø, Ø, Ø, Ø>
h_2 = <Sunny, Warm, Normal, Strong, Warm, Same>
h_3 = <Sunny, Warm, ?, Strong, Warm, Same>
h_4 = <Sunny, Warm, ?, Strong, Warm, Same>
h_5 = <Sunny, Warm, ?, Strong, ?, ?>

- **Shortcomings of *Find-S***
  - Can't tell whether it has learned concept
  - Can't tell when training data inconsistent
  - Picks a maximally specific *h* (why?)
  - Depending on *H*, there might be several!

Version Spaces

- **Definition: Consistent Hypotheses**
  - **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 <*x, c(x)*> in *D*.**
  - $\textit{Consistent}\ (h, D) \equiv \forall <x, c(x)> \in D\ .\ h(x) = c(x)$
- **Definition: Version Space**
  - **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*.**
  - $VS_{H,D} \equiv \{ h \in H \mid \textit{Consistent}\ (h, D) \}$

Candidate Elimination Algorithm [1]

1. Initialization

$G \leftarrow$ (singleton) set containing most general hypothesis in *H*, denoted {<?, … , ?>}

$S \leftarrow$ set of most specific hypotheses in *H*, denoted {<Ø, … , Ø>}

2. For each training example *d*

If *d* is a positive example (*Update-S*)

Remove from *G* any hypotheses inconsistent with *d*

For each hypothesis *s* in *S* that is not consistent with *d*

Remove *s* from *S*

Add to *S* all minimal generalizations *h* of *s* such that

1. *h* is consistent with *d*
2. Some member of *G* is more general than *h*

(These are the greatest lower bounds, or *meets*, $s \vee d$, in $VS_{H,D}$)

Remove from *S* any hypothesis that is more general than another hypothesis in *S* (remove any dominated elements)

Candidate Elimination Algorithm [2]

(continued)

If *d* is a negative example (*Update-G*)

Remove from *S* any hypotheses inconsistent with *d*

For each hypothesis *g* in *G* that is not consistent with *d*

Remove *g* from *G*

Add to *G* all minimal specializations *h* of *g* such that

1. *h* is consistent with *d*
2. Some member of *S* is more specific than *h*

(These are the least upper bounds, or *joins*, $g \wedge d$, in $VS_{H,D}$)

Remove from *G* any hypothesis that is less general than another hypothesis in *G* (remove any dominating elements)

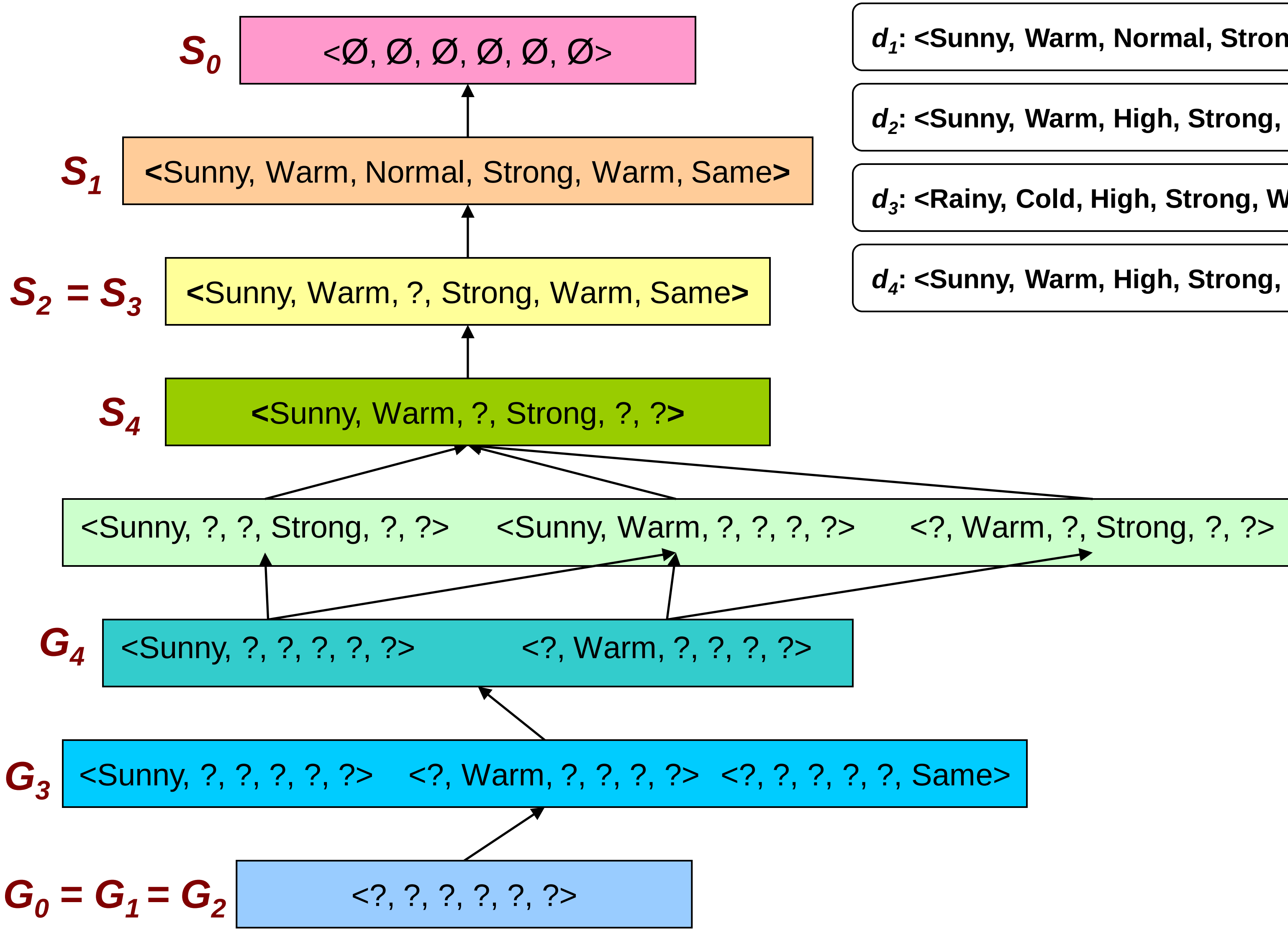
Example Trace

d_1: **<Sunny, Warm, Normal, Strong, Warm, Same, <u>Yes</u>>**

d_2: **<Sunny, Warm, High, Strong, Warm, Same, <u>Yes</u>>**

d_3: **<Rainy, Cold, High, Strong, Warm, Change, <u>No</u>>**

d_4: **<Sunny, Warm, High, Strong, Cool, Change, <u>Yes</u>>**



Candidate-Elimination Algorithm

- **The choice of training examples**
  - **The least query times: $\log_2 |VS|$**
- **How can partially learned concepts be used?**
  - **Possible to classify certain examples**
    - **Positive: satisfying every member of S**
    - **Negative: satisfying none member of G**
    - **Uncertain: see the proportion of hypotheses voting positive**

Inductive Bias

- **Fundamental assumption of inductive learning:**
  - **The 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 Bias

- **Fundamental questions:**
  - **What if the target concept is not contained in hypothesis space?**
  - **The relationship between the size of hypothesis space, the ability of algorithm to generalize to unobserved instances, the number of training examples that must be observed**

Inductive Bias

It can't be represented in H we defined

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Rainy	Warm	Normal	Strong	Warm	Same	No
3	Cloudy	Warm	Normal	Strong	Warm	Same	Yes

Inductive Bias

Fundamental property of inductive inference

A learner that makes no a priori assumptions regarding the identity of the target concept has no rational basis for classifying any unseen instances

Inductive bias

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

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

Terminology

- **Supervised Learning**
  - **Concept - function from observations to categories (so far, boolean-valued: +/-)**
  - **Target (function) - true function *f***
  - **Hypothesis - proposed function *h* believed to be similar to *f***
  - **Hypothesis space - space of all hypotheses that can be generated by the learning system**
  - **Example - tuples of the form <*x, f(x)*>**
  - **Instance space (*aka* example space) - space of all possible examples**
  - **Classifier - discrete-valued function whose range is a set of class labels**
- **The Version Space Algorithm**
  - **Algorithms: *Find-S*, *List-Then-Eliminate*, candidate elimination**
  - **Consistent hypothesis - one that correctly predicts observed examples**
  - **Version space - space of all currently consistent (or *satisfiable*) hypotheses**
- **Inductive Learning**
  - **Inductive generalization - process of generating hypotheses that describe cases not yet observed**
  - **The inductive learning hypothesis**