CMT311 Principles of Machine Learning

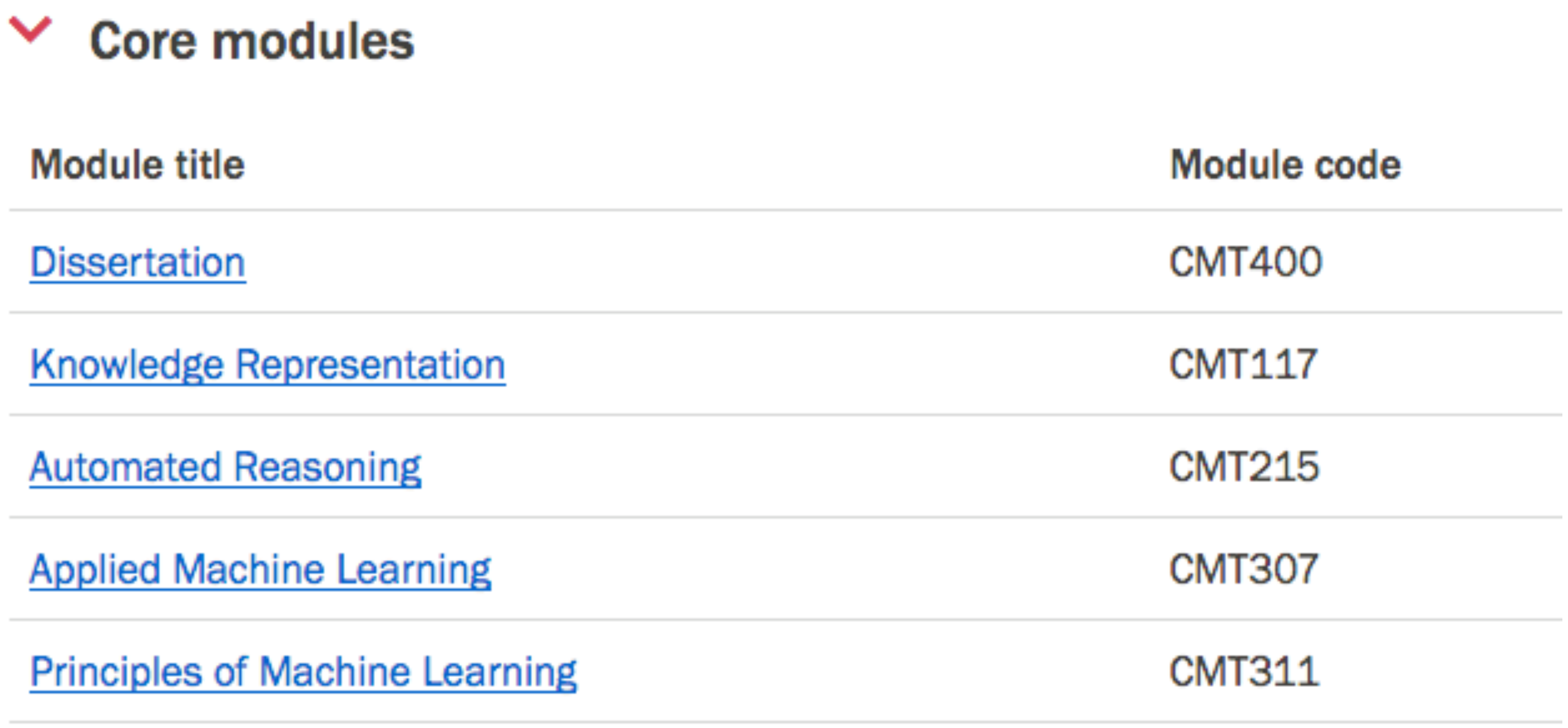
Introduction & Concept Learning

Angelika Kimmig
KimmigA@cardiff.ac.uk

04.10.2019

Cardiff MSc AI

Artificial Intelligence is more than Machine Learning, Machine Learning is more than Deep Learning



Core modules

Module title	Module code
Dissertation	CMT400
Knowledge Representation	CMT117
Automated Reasoning	CMT215
Applied Machine Learning	CMT307
Principles of Machine Learning	CMT311

cf also e.g.

- Adnan Darwiche's "Human-Level Intelligence or Animal-Like Abilities?"
- Hector Levesque's "Common sense, the Turing test, and the quest for real AI"

(the library provides online access to both)

CMT311 Topics

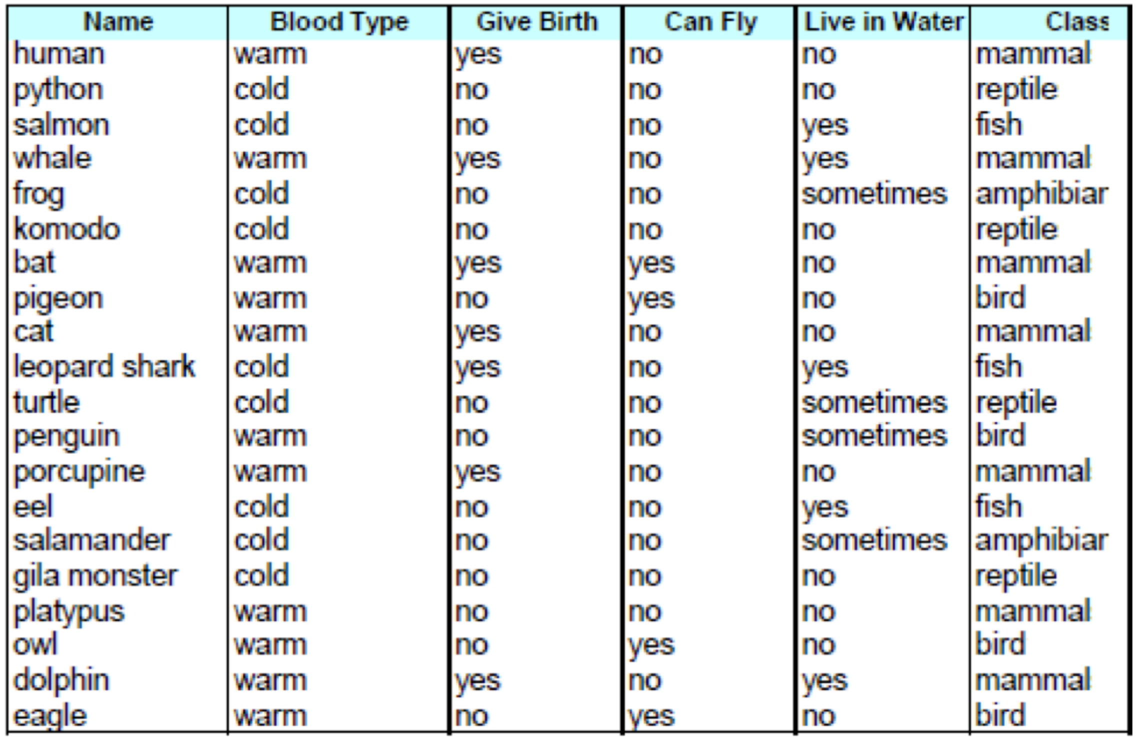
- Learning Theory
- Logic & Learning
- Probabilistic Graphical Models
- Statistical Relational Learning

Learning Theory

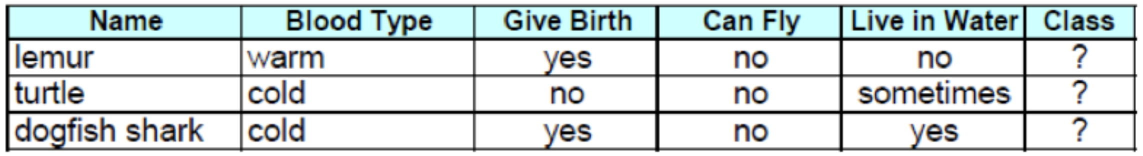
- Formal study of fundamental questions such as
  - What is (machine) learning?
  - What is needed for machine learning to succeed?
  - How can we measure success/quality?
  - …

Logic & Learning

- Concept Learning
- Rule Learning
- Knowledge in Learning



Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
human	warm	yes	no	no	mammal
python	cold	no	no	no	reptile
salmon	cold	no	no	yes	fish
whale	warm	yes	no	yes	mammal
frog	cold	no	no	sometimes	amphibiar
komodo	cold	no	no	no	reptile
bat	warm	yes	yes	no	mammal
pigeon	warm	no	yes	no	bird
cat	warm	yes	no	no	mammal
leopard shark	cold	yes	no	yes	fish
turtle	cold	no	no	sometimes	reptile
penguin	warm	no	no	sometimes	bird
porcupine	warm	yes	no	no	mammal
eel	cold	no	no	yes	fish
salamander	cold	no	no	sometimes	amphibiar
gila monster	cold	no	no	no	reptile
platypus	warm	no	no	no	mammal
owl	warm	no	yes	no	bird
dolphin	warm	yes	no	yes	mammal
eagle	warm	no	yes	no	bird



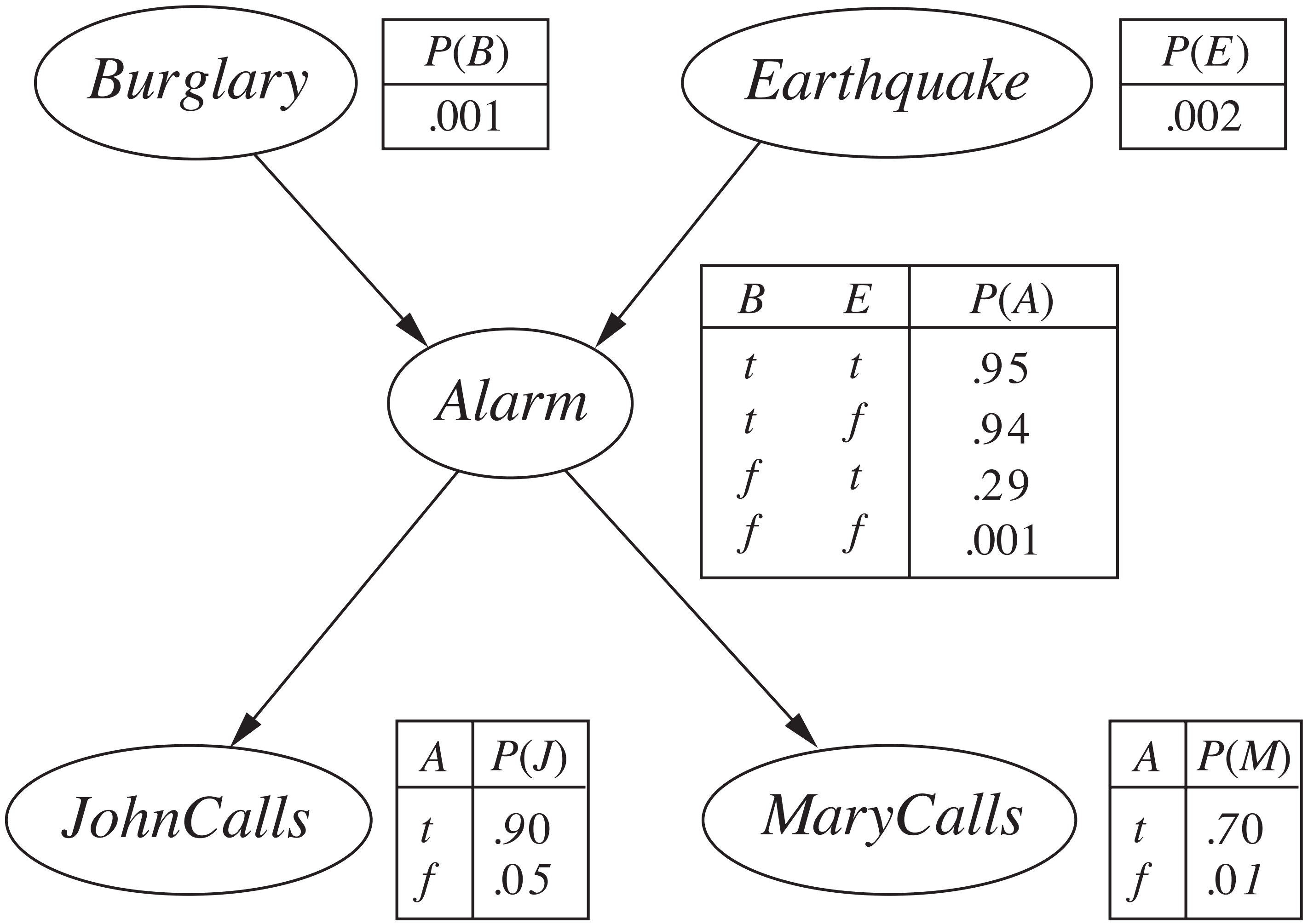
Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
lemur	warm	yes	no	no	?
turtle	cold	no	no	sometimes	?
dogfish shark	cold	yes	no	yes	?

R1: if (Give Birth = no) & (Can Fly = yes), then bird
R2: if (Give Birth = no) & (Live in Water = yes), then fish
R3: if (Give Birth = yes) & (Blood Type = warm), then mammal
R4: if (Give Birth = no) & (Can Fly = no), then reptile
R5: if (Live in Water = sometimes), then amphibian

Probabilistic Graphical Models

- What if the world is not black and white?
- How to reason?
- How to learn models?
- How to deal with relational information?

statistical relational learning



$P(B)$
.001

$P(E)$
.002

B	E	$P(A)$
t	t	.95
t	f	.94
f	t	.29
f	f	.001

A	$P(J)$
t	.90
f	.05

A	$P(M)$
t	.70
f	.01

[Figure: Russell & Norvig, 2009]

Textbooks

main

- Shai Shalev-Shwartz and Shai Ben-David.
  Understanding Machine Learning: From Theory to Algorithms.
  Cambridge University Press, 2014.
  Library has copies & provides online access.
  Free pdf at http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning

- David Barber. Bayesian Reasoning and Machine Learning.
  Cambridge University Press, 2012.
  Library has copies.
  Free pdf at http://www.cs.ucl.ac.uk/staff/d.barber/brml/

- Stuart Russell and Peter Norvig.
  Artificial Intelligence: A Modern Approach.
  3rd ed. Pearson, 2009.

- Tom Mitchell.
  Machine Learning.
  McGraw-Hill, 1997.

Note: Shai Ben-David's youtube playlist (Machine Learning Theory)

- great lecture videos
- **but:** very slow, and much more detail than we'll cover, both in terms of breadth (entire book) and depth (full details of proofs)

General Information

- module runs over both semesters; lecturer for second semester to be determined
- timetable suggests clear split between lectures & practicals, but we'll mix these freely
    - actively engaging with material is crucial for success!
- assessment: 30% coursework, 70% written exam

Today

- Introduction
  - What is (machine) learning?
  - Why machine learning?
  - Types of machine learning
  - How to specify a learning task?
- Example setting: concept learning

What is (machine) learning?

- Learning is the process of converting **experience** into **expertise** or knowledge. [Shalev-Shwartz & Ben-Davis]

- A computer is said to learn from **experience** E with respect to some **task** T and **performance measure** P, if its performance at tasks in T, as measured by P, **improves** with experience E. [Mitchell]

- memorisation vs generalisation / inductive reasoning

- common sense & prior knowledge: inductive bias

Why machine learning?

- Tasks that are **too complex** to program
  - within human capabilities
  - beyond human capabilities
- **Adaptivity**

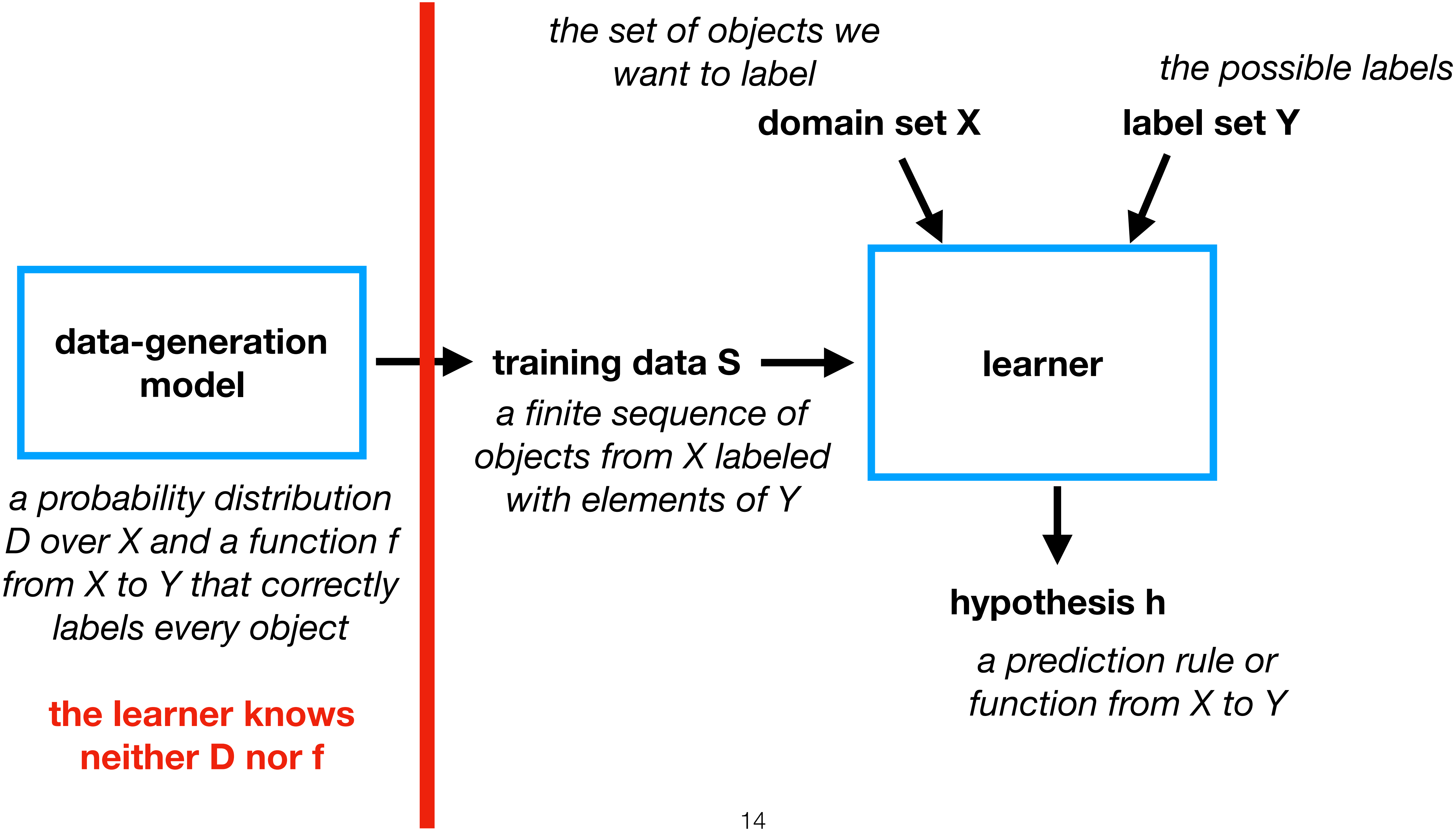
Types of learning

- supervised vs unsupervised
  - reinforcement learning
- active vs passive learners
- helpfulness of the teacher
  - helpful / random / adversarial
- online vs batch

most of this module

How to specify learning tasks?

The Statistical Learning Framework



data-generation model

a probability distribution D over X and a function f from X to Y that correctly labels every object

the learner knows neither D nor f

training data S

a finite sequence of objects from X labeled with elements of Y

the set of objects we want to label

domain set X

the possible labels

label set Y

learner

hypothesis h

a prediction rule or function from X to Y

Measure of success

- **error** of a hypothesis h = probability of h assigning a wrong label to a random object x drawn from D
- formally:

$$L_{D,f}(h) = D(\{x \in X \mid h(x) \neq f(x)\})$$

error (or loss) of hypothesis h with respect to distribution D and correct labeling function f

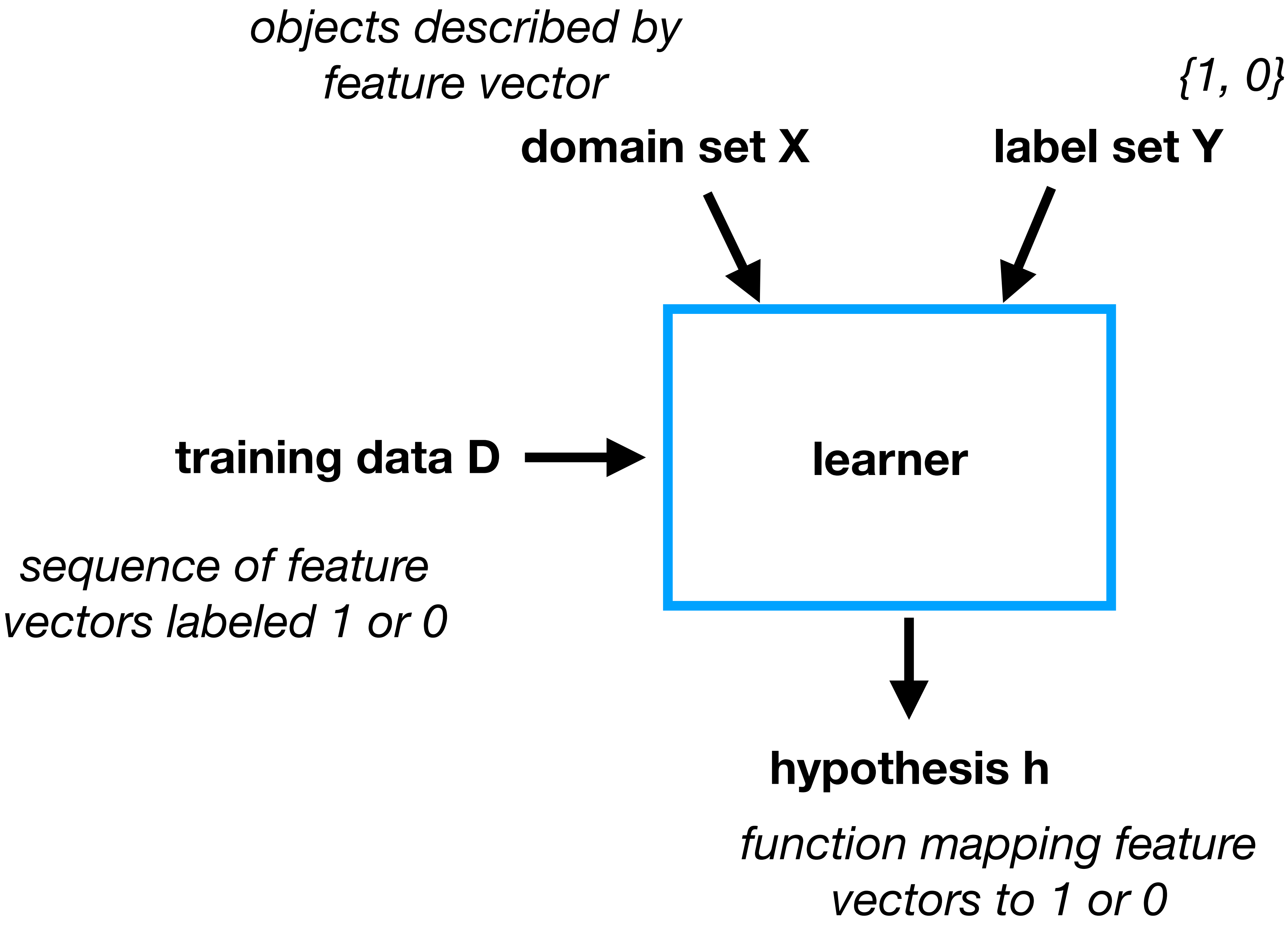
probability according to distribution D of the subset of X where hypothesis h and correct function f disagree

Challenge

- Given a learning task, we want to **build a learner with low error**, but the error we just saw **depends on the unknown distribution D and function f** — how can we do this? Is it even possible to do this?
- We'll study these abstract questions in the next weeks
    - focus on key ideas and principles
    - details of formal proofs are less important

Example setting: Concept learning

Boolean Concept Learning



objects described by feature vector

domain set X

{1, 0}

label set Y

training data D

sequence of feature vectors labeled 1 or 0

learner

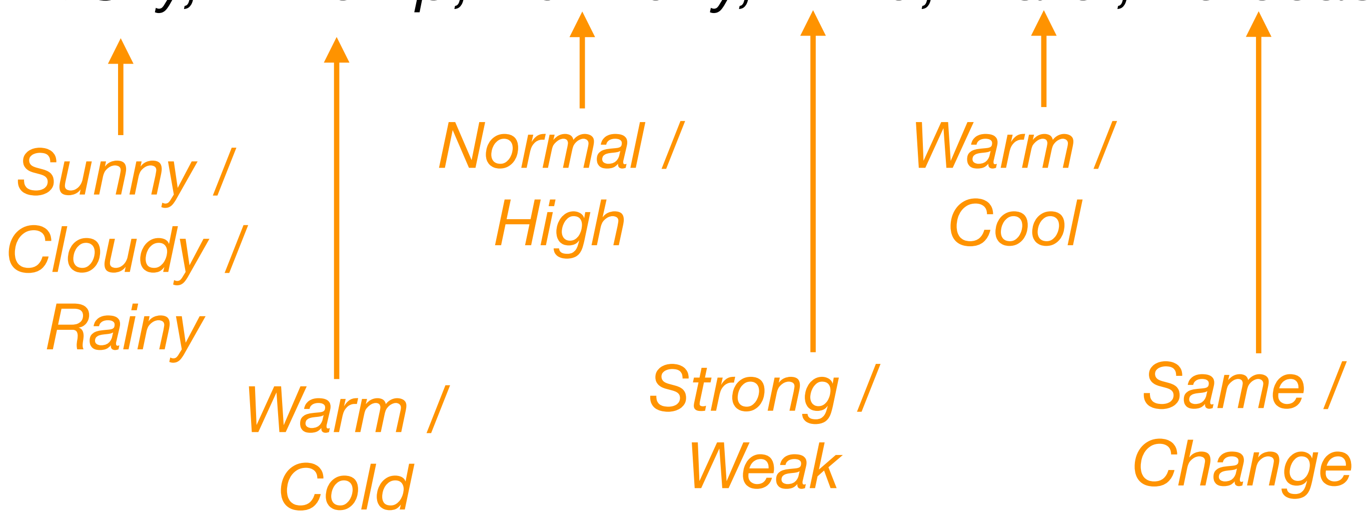
hypothesis h

function mapping feature vectors to 1 or 0

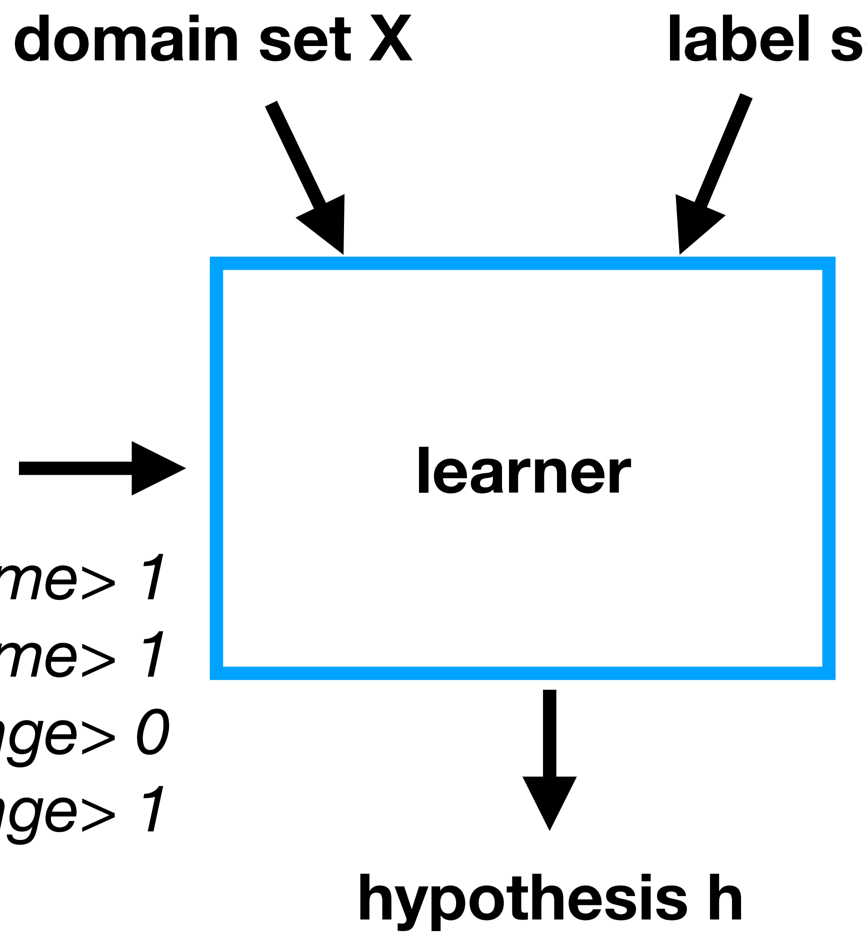
Example

days described by

< Sky, AirTemp, Humidity, Wind, Water, Forecast >



labels 1 / 0 indicating whether someone enjoys their favourite sport on that day



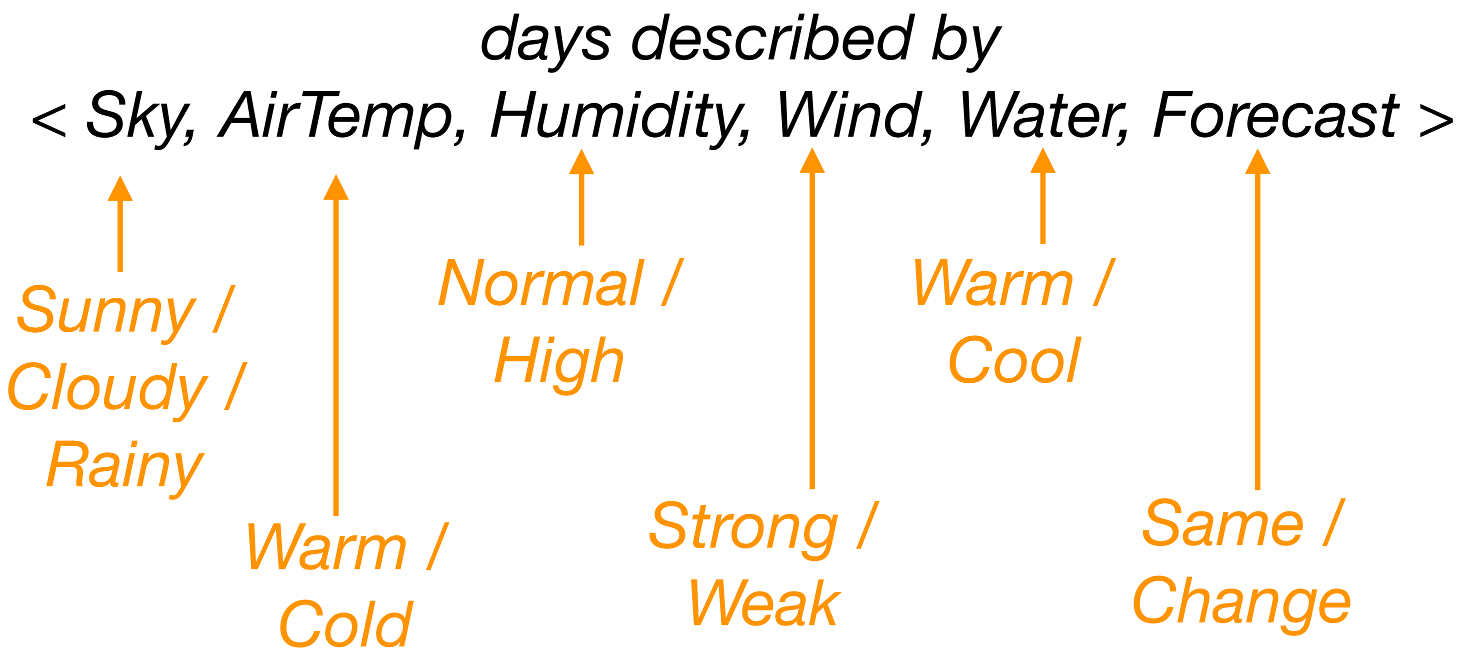
training data D

<Sunny, Warm, Normal, Strong, Warm, Same> 1
<Sunny, Warm, High, Strong, Warm, Same> 1
<Rainy, Cold, High, Strong, Warm, Change> 0
<Sunny, Warm, High, Strong, Cool, Change> 1

Example: Hypothesis Space

- hypothesis = conjunction of constraints on features
  - any value acceptable (“?”)
  - only one given value acceptable
  - no value acceptable (“-“)
- if instance x satisfies all constraints of hypothesis h, then h(x) = 1, otherwise, h(x) = 0

Example



days described by

< Sky, AirTemp, Humidity, Wind, Water, Forecast >

Sunny / Cloudy / Rainy

Warm / Cold

Normal / High

Strong / Weak

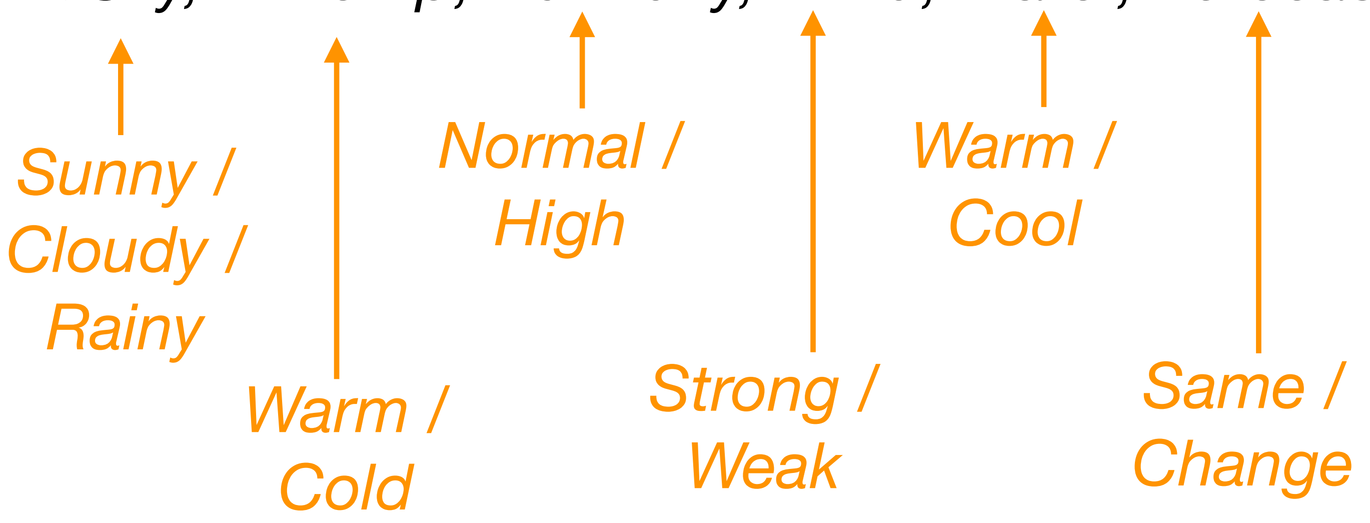
Warm / Cool

Same / Change

- <Sunny,Warm,Normal,Strong,Warm,Same>
- <?,Cold,High,?,?,?>
- <?,?,?,?,?,?>
- <-,-,-,-,-,->
- <Sunny,?,?,-,Warm,Same>

Example

days described by
< Sky, AirTemp, Humidity, Wind, Water, Forecast >



Sunny / Cloudy / Rainy

Warm / Cold

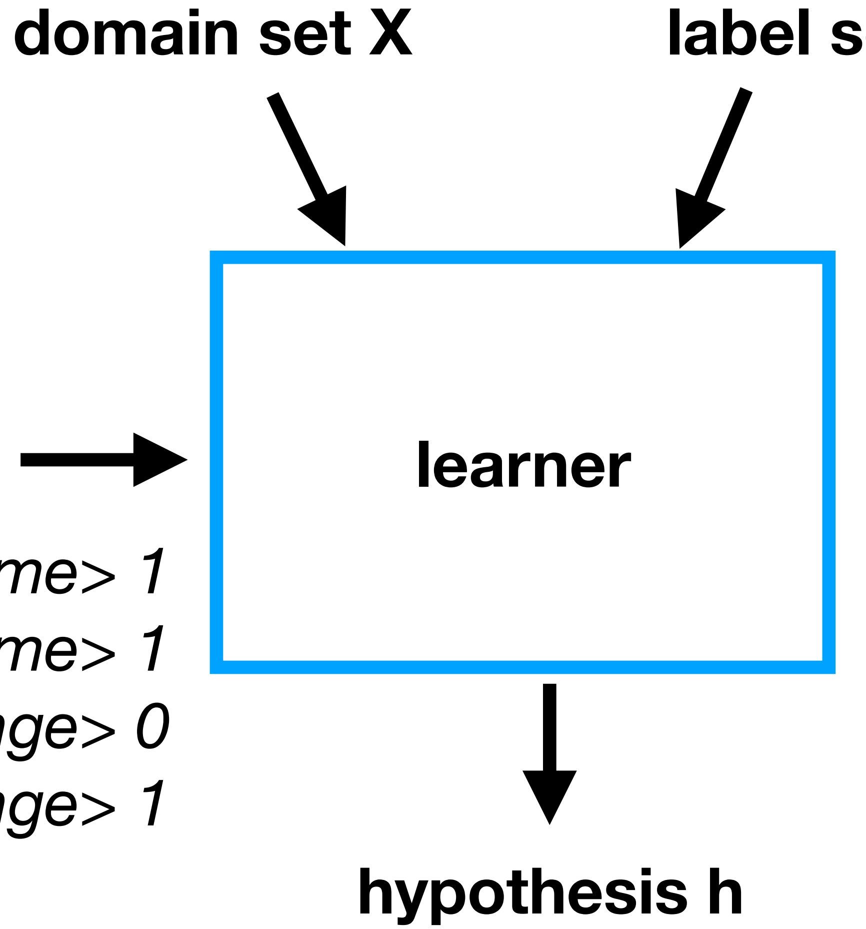
Normal / High

Strong / Weak

Warm / Cool

Same / Change

labels 1 / 0 indicating whether someone enjoys their favourite sport on that day



domain set X

label set Y

training data D

learner

hypothesis h

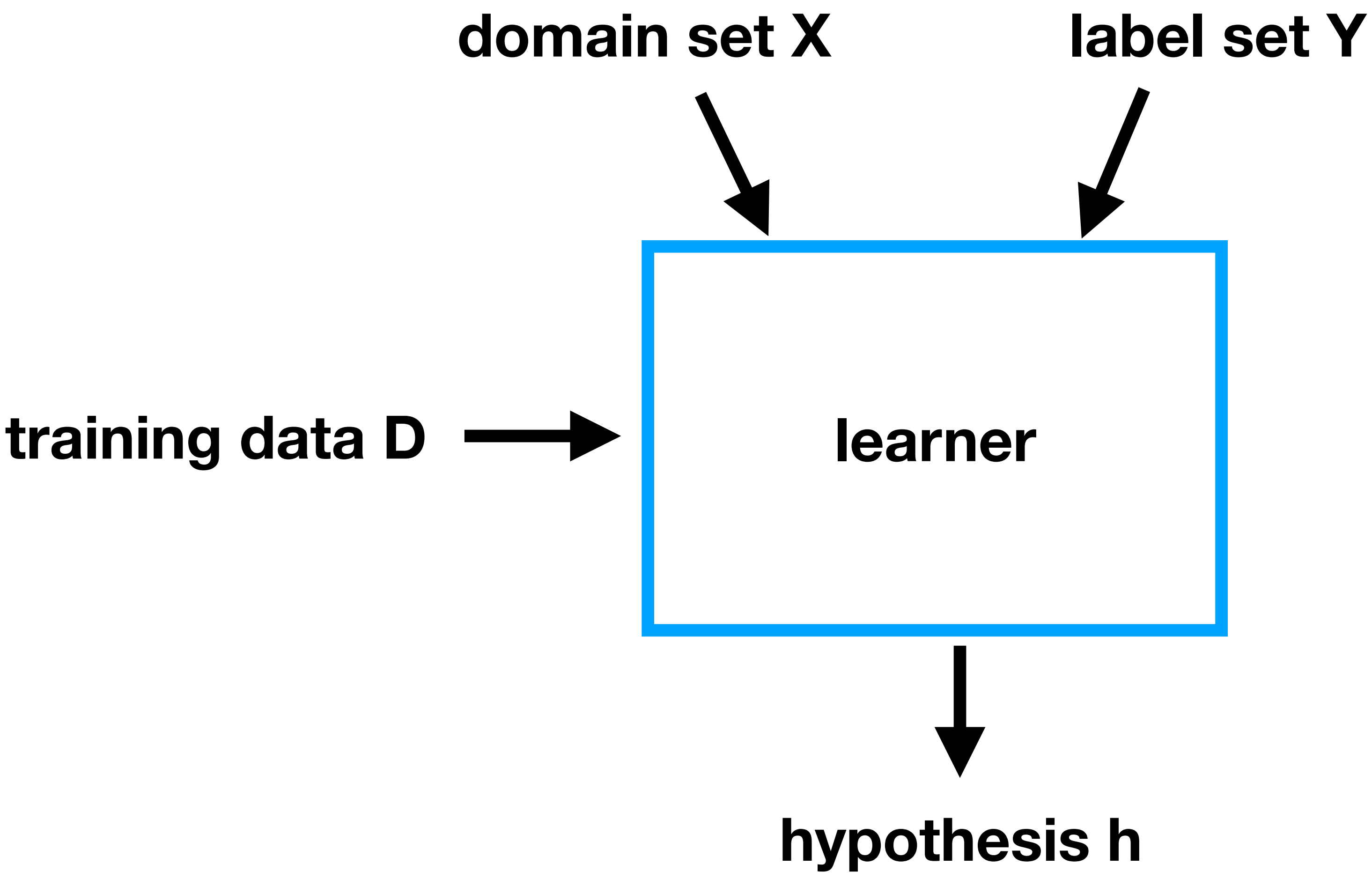
<Sunny, Warm, Normal, Strong, Warm, Same> 1
<Sunny, Warm, High, Strong, Warm, Same> 1
<Rainy, Cold, High, Strong, Warm, Change> 0
<Sunny, Warm, High, Strong, Cool, Change> 1

conjunction of attribute constraints

Inductive learning hypothesis

- Any hypothesis that performs well on a **sufficiently large training set** D will also perform well on the **full set** X

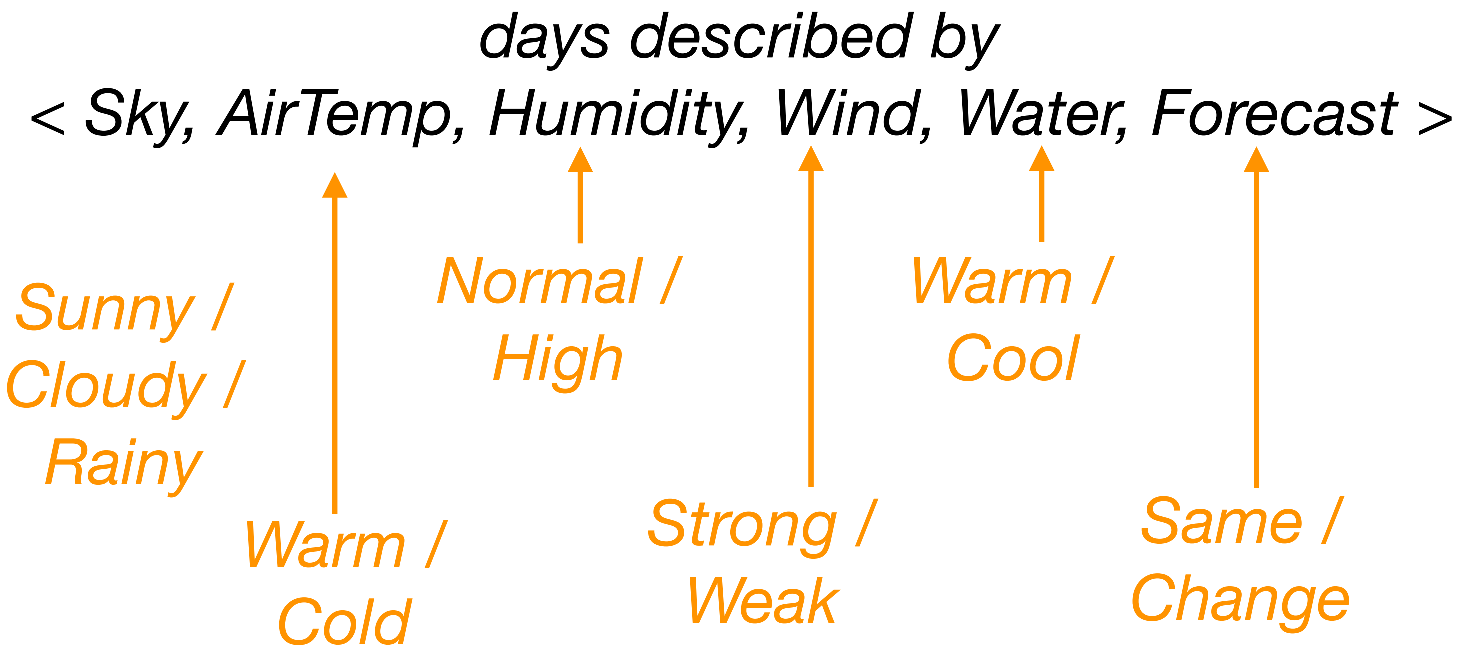
- Thus, once we fixed a hypothesis space H, **learning** equals **searching** for a good h in H



Learning as Search

- In our example, the hypothesis space H is finite (why?)
- naive learner = enumerate & test all hypotheses in H
- In practice, H is often much larger or infinite, so learner needs to be “smarter”
- Key trick: exploit general-to-specific order on H

General-to-specific ordering



days described by

< Sky, AirTemp, Humidity, Wind, Water, Forecast >

Sunny / Cloudy / Rainy

Warm / Cold

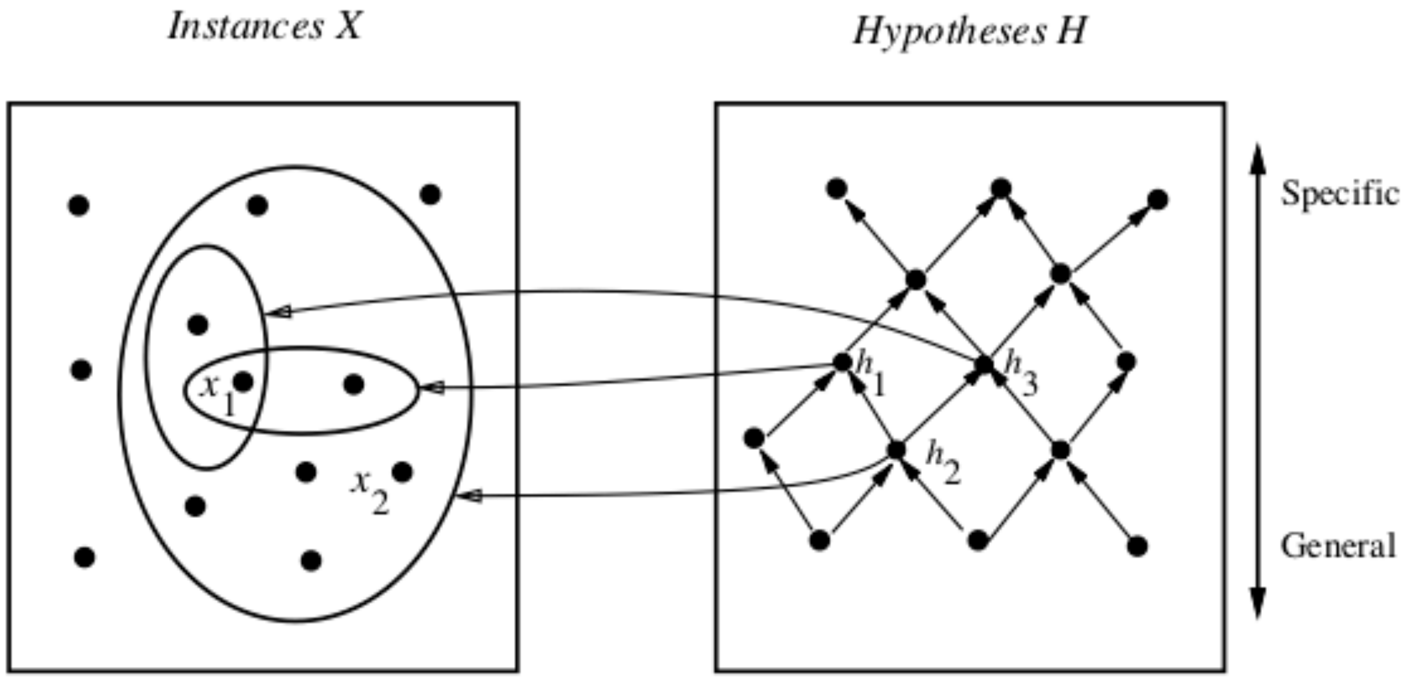
Normal / High

Strong / Weak

Warm / Cool

Same / Change

- <?,Cold,High,?,?,?>
- <?,Cold,?,?,?,?>
- <?,?,High,?,?,?>
- <?,?,?,?,?,?>



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, ?*>

[Figure: Mitchell]

Formal definition

- Let h_j and h_k be two Boolean-valued functions defined over X.

<?,Cold,?,?,?,?> $\geq_g$<?,Cold,High,?,?,?>

- Then h_j is **more general than or equal to** h_k**,** $h_j \geq_g h_k$, if and only if $\forall x \in X : h_k(x) = 1 \rightarrow h_j(x) = 1$
- h_j is **strictly more general than** h_k, $h_j >_g h_k$, if and only if $h_j \geq_g h_k$ and $h_k \not\geq_g h_j$
- h_j is **more specific than** h_k if and only if h_k is more general than h_j
- note: these notions are **independent** of the target concept

Exercise

- Consider Boolean concept learning with the object set X containing all points (x,y) on the square grid with coordinates from 0 to 10, and hypotheses of the form ($a \le x \le b \land c \le y \le d$) with a,b,c,d integers in [0,10]

- What is the most general hypothesis in this space, and what is the most specific one?

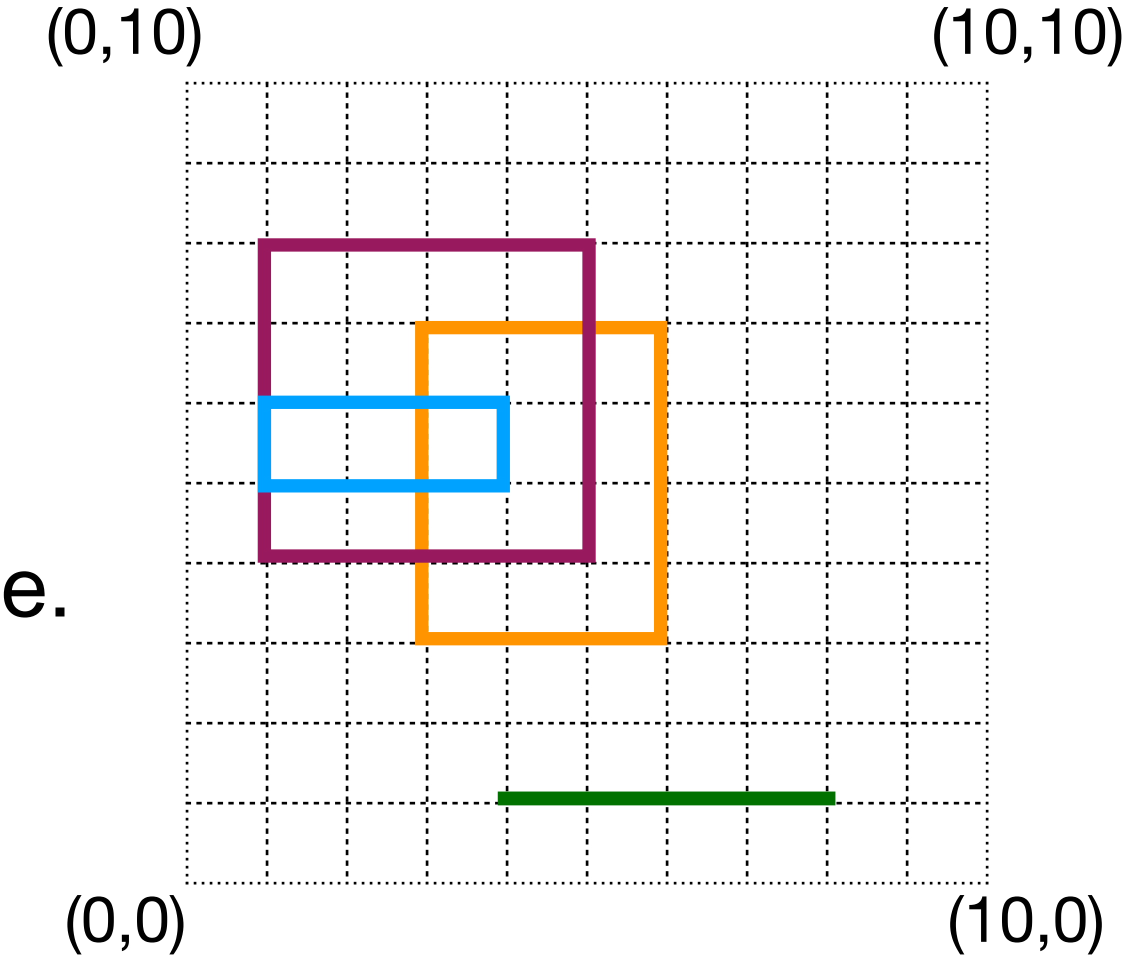
- Give a graphical interpretation of the “more general than” order for this space.

($1 \le x \le 4 \land 5 \le y \le 6$)

($3 \le x \le 6 \land 3 \le y \le 7$)

($1 \le x \le 5 \land 4 \le y \le 8$)

($4 \le x \le 8 \land 1 \le y \le 1$)



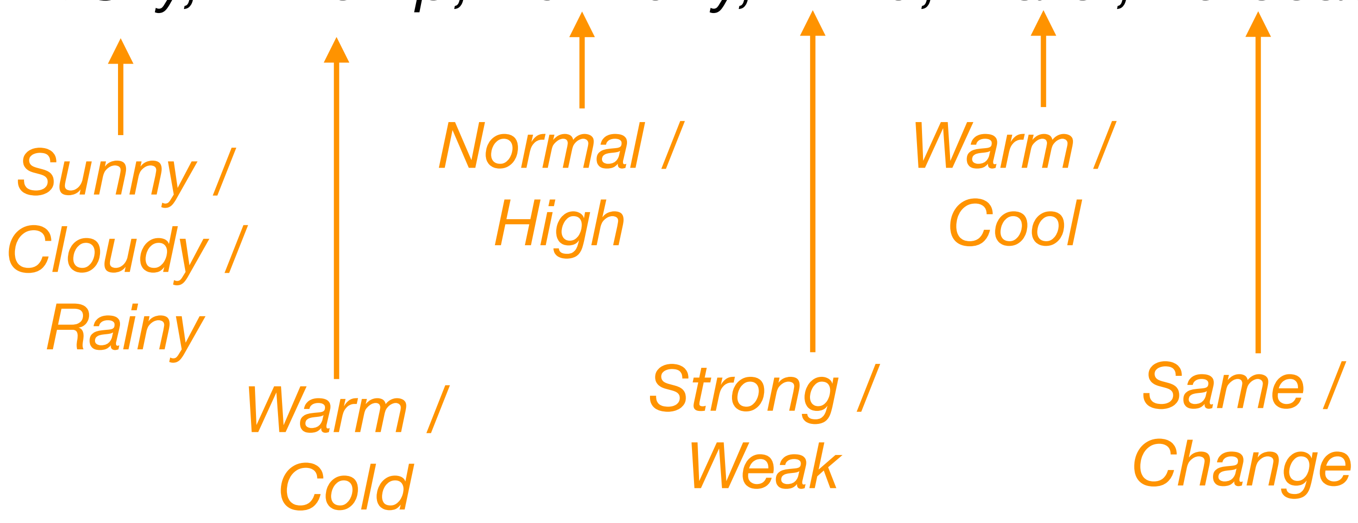
A basic learner: FIND-S

- set *h* to the most specific hypothesis in *H*
- for each positive *x* in *D*
  - for each constraint *a* in *h*
    - if *x* does not satisfy *a* then replace *a* in *h* by the next more general constraint *a’* that is satisfied by *x*
- return *h*

Example

days described by

< Sky, AirTemp, Humidity, Wind, Water, Forecast >



Sunny / Cloudy / Rainy

Warm / Cold

Normal / High

Strong / Weak

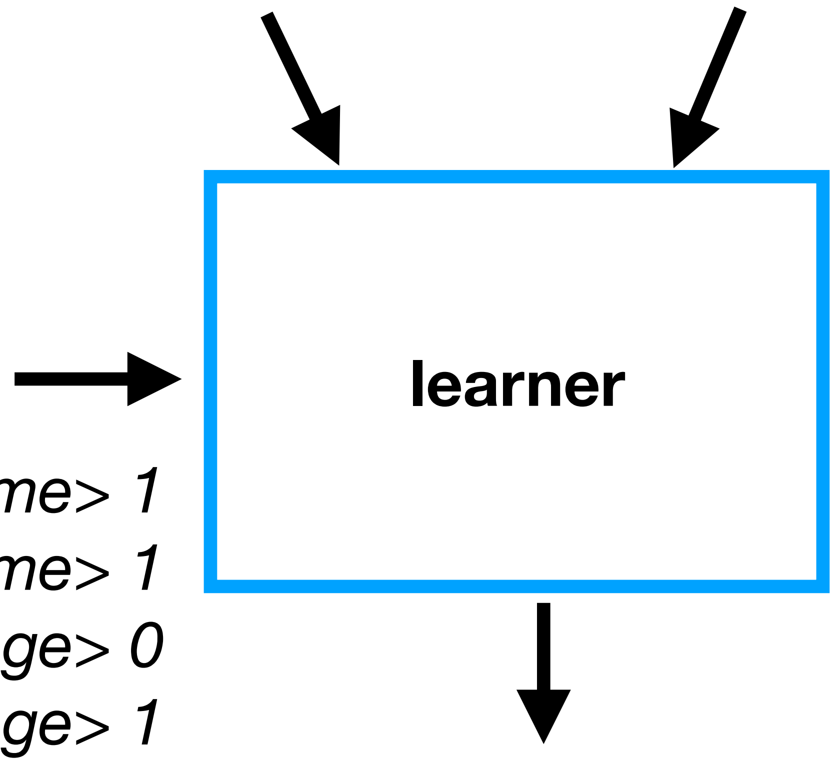
Warm / Cool

Same / Change

labels 1 / 0 indicating whether someone enjoys their favourite sport on that day

domain set X

label set Y



training data D

learner

<Sunny, Warm, Normal, Strong, Warm, Same> 1
<Sunny, Warm, High, Strong, Warm, Same> 1
<Rainy, Cold, High, Strong, Warm, Change> 0
<Sunny, Warm, High, Strong, Cool, Change> 1

hypothesis h

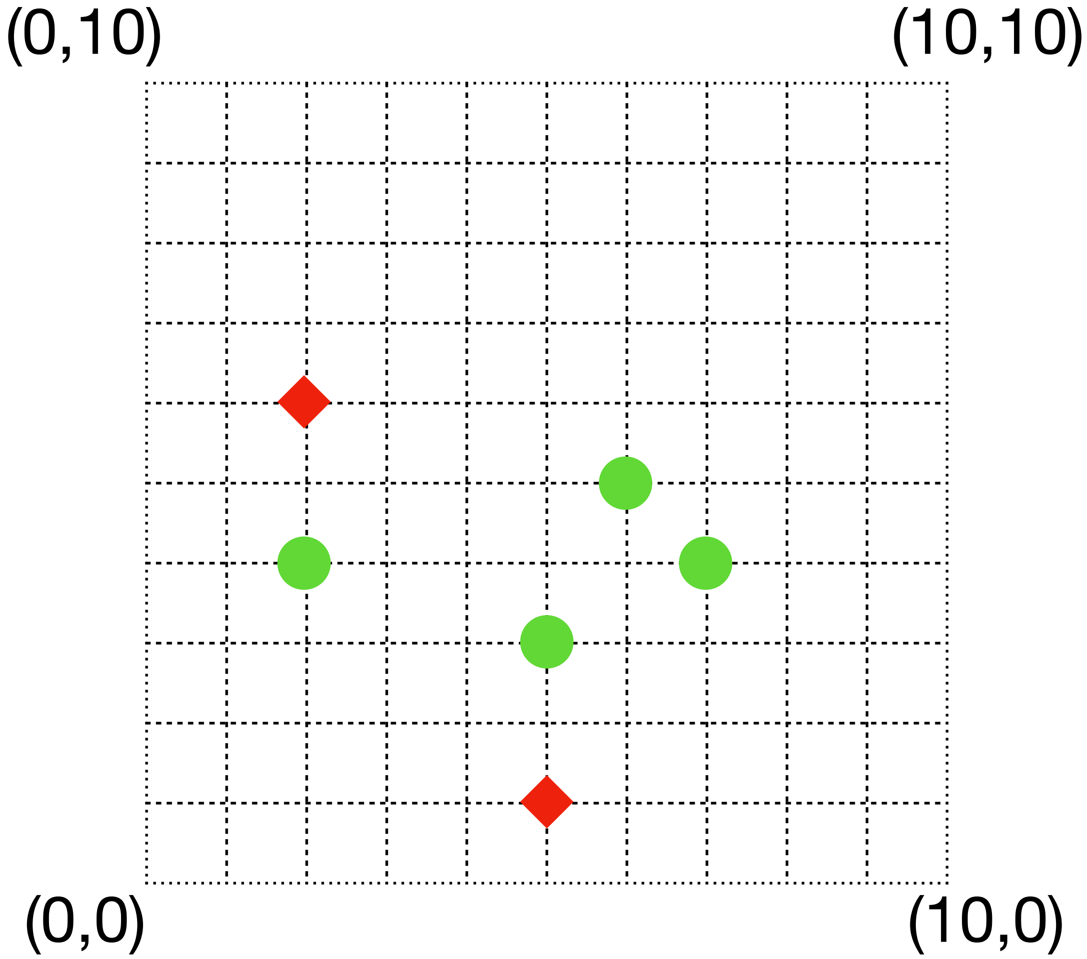
conjunction of attribute constraints

Exercise

- Consider again the space of rectangles (a≤x≤b ∧ c≤y≤d) on the [0,10]x[0,10] grid.

- Trace the FIND-S algorithm for the following sequence of examples:

(2,4)	1
(7,4)	1
(5,1)	0
(5,3)	1
(2,6)	0
(6,5)	1



FIND-S: Discussion

- Is it safe to ignore the negative examples?
- Yes, as long as the **training data** has been **correctly labeled** by a function **f** that is **in H**
- These two conditions make it impossible for FIND-S to generalise its hypothesis too much

FIND-S: Discussion

- the hypothesis returned by FIND-S is
    - the most specific one in H that correctly labels all positive training examples
    - correctly labels all negative training examples, provided that the correct target concept is in H and the training data is correct
- open questions:
    - has the learner converged to the correct answer?
    - why prefer the most specific h?
    - what if the training data is not labeled correctly?
    - what if there are several maximally specific hypotheses for the training data?

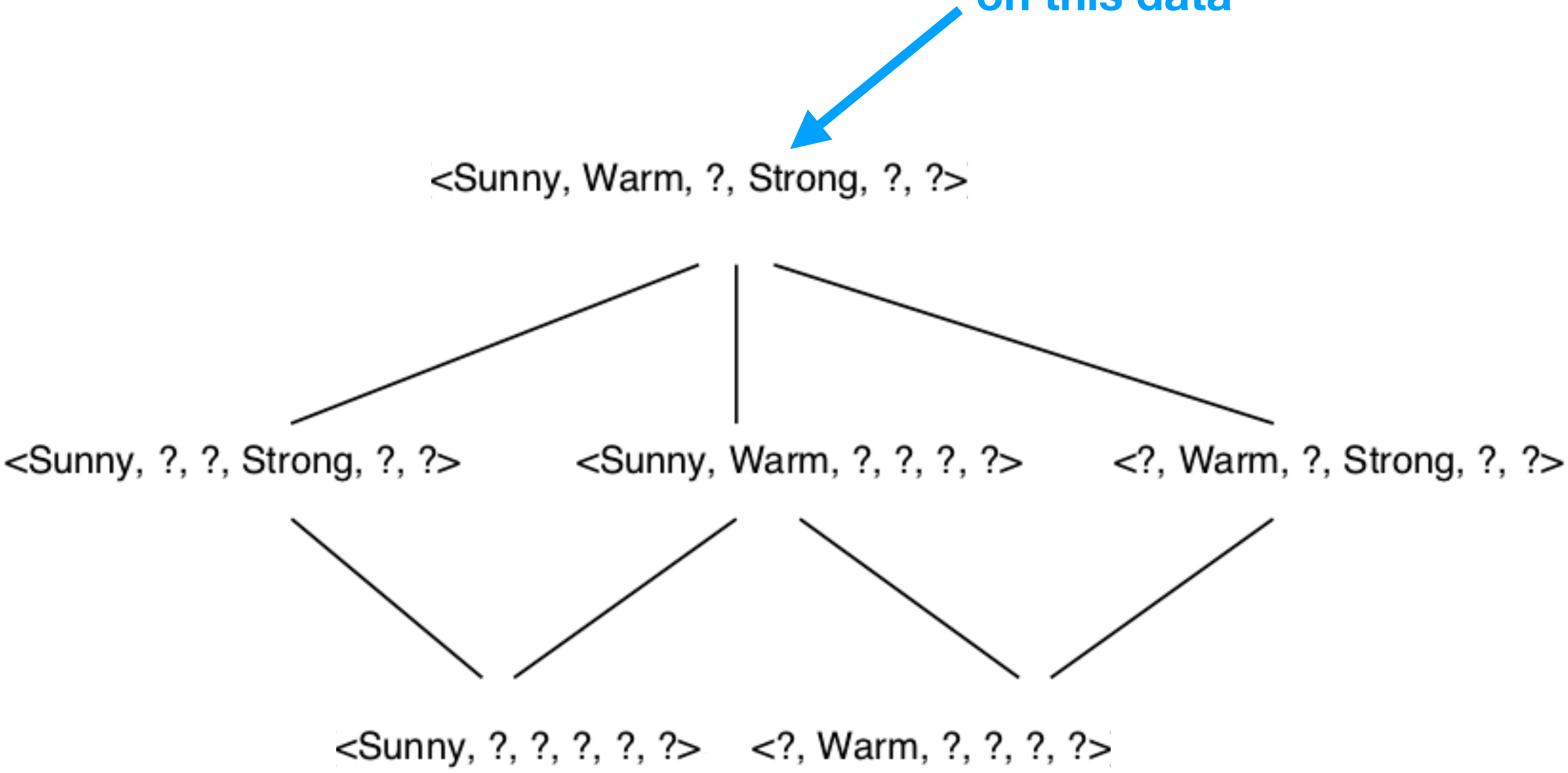
Using version spaces

- A hypothesis h is **consistent** with training data D if and only if for all examples (x,y) in D, h(x)=y
- Goal: a learner that finds all hypotheses in H that are consistent with D, using the “more general than” order
- The **version space** $VS_{H,D}$ with respect to hypothesis space H and training data D is the set of all hypotheses in H consistent with D

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

Example

<Sunny, Warm, Normal, Strong, Warm, Same> 1
<Sunny, Warm, High, Strong, Warm, Same> 1
<Rainy, Cold, High, Strong, Warm, Change> 0
<Sunny, Warm, High, Strong, Cool, Change> 1


the hypothesis returned by FIND-S on this data
<Sunny, Warm, ?, Strong, ?, ?>
<Sunny, ?, ?, Strong, ?, ?>
<Sunny, Warm, ?, ?, ?, ?>
<?, Warm, ?, Strong, ?, ?>
<Sunny, ?, ?, ?, ?, ?>
<?, Warm, ?, ?, ?, ?>

[Figure: Mitchell]

another learner: LIST-THEN-ELIMINATE

- VS = list of all hypotheses in H
- for each example (x,y) in D
  - remove from VS all h with h(x)≠y
- return VS

Version space boundaries

- The **general boundary G** with respect to hypothesis space H and training data D is the set of maximally general members of H consistent with D.

$$G \equiv \{g \in H \mid consistent(g, D) \wedge \neg \exists g' \in H : g' >_g g \wedge consistent(g', D)\}$$

- The **specific boundary S** with respect to hypothesis space H and training data D is the set of minimally general members of H consistent with D.

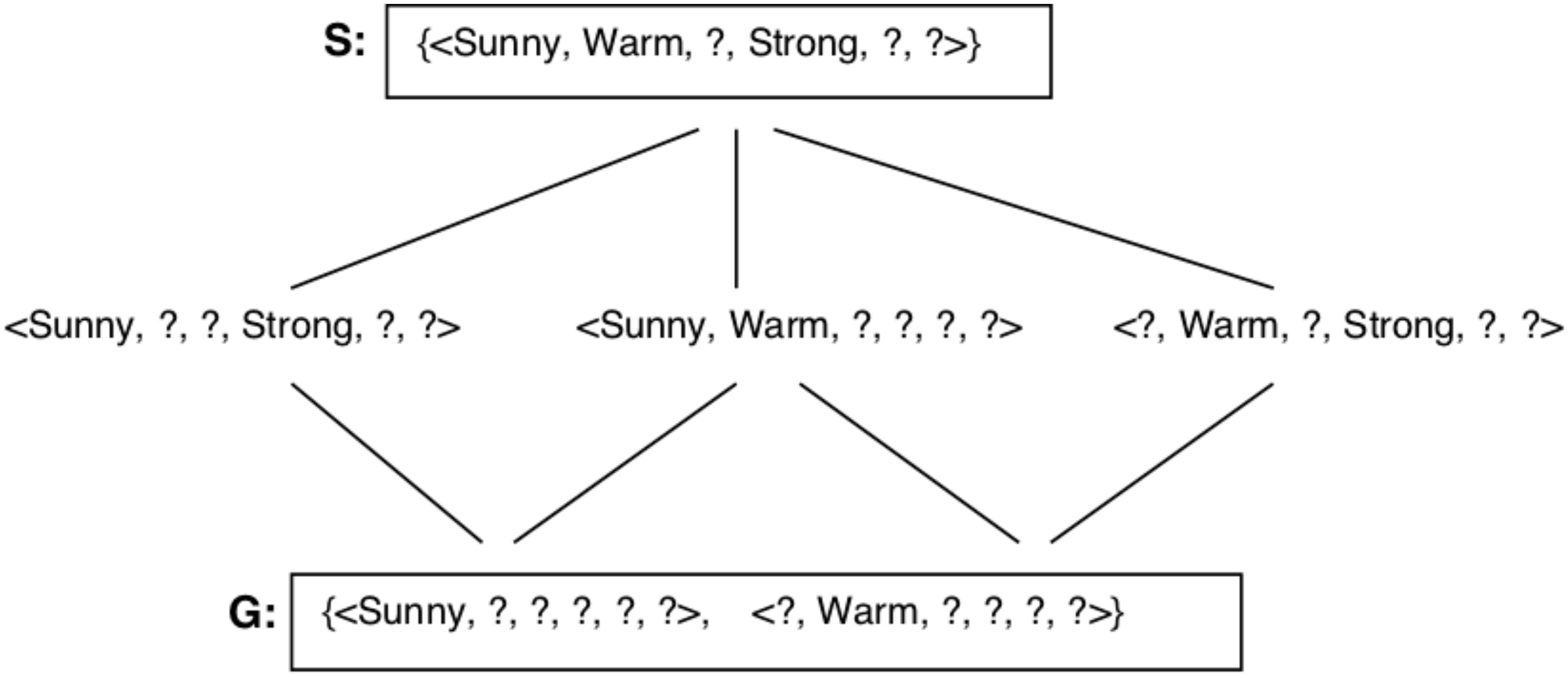
$$S \equiv \{s \in H \mid consistent(s, D) \wedge \neg \exists s' \in H : s >_g s' \wedge consistent(s', D)\}$$

- Every member of the version space lies between G and S:

$$VS_{H,D} = \{h \in H \mid \exists s \in S : \exists g \in G : g \geq_g h \geq_g s\}$$

Example

<Sunny, Warm, Normal, Strong, Warm, Same> 1
<Sunny, Warm, High, Strong, Warm, Same> 1
<Rainy, Cold, High, Strong, Warm, Change> 0
<Sunny, Warm, High, Strong, Cool, Change> 1



S: {<Sunny, Warm, ?, Strong, ?, ?>}

<Sunny, ?, ?, Strong, ?, ?>   <Sunny, Warm, ?, ?, ?, ?>   <?, Warm, ?, Strong, ?, ?>

G: {<Sunny, ?, ?, ?, ?, ?>, <?, Warm, ?, ?, ?, ?>}

[Figure: Mitchell]

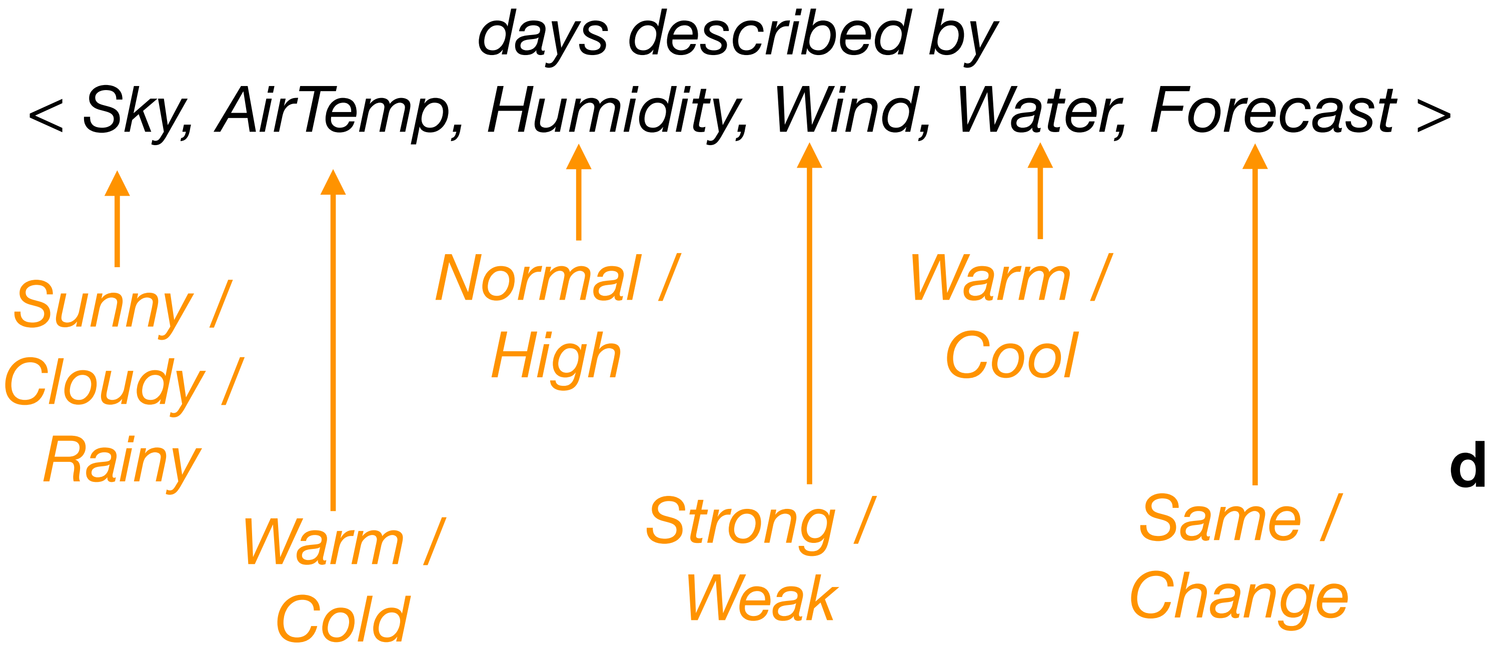
CANDIDATE-ELIMINATION

- G = set of maximally general hypotheses in H
- S = set of maximally specific hypotheses in H
- for each training example d
    - if d is positive
        - remove from G any h inconsistent with d
        - for each s in S that is not consistent with d
            - remove s from S
            - add to S all minimal generalisations h of s such that h is consistent with d and some member of G is more general than h
            - remove from S any h that is more general than some h’ in S
    - if d is negative
        - remove from S any h inconsistent with d
        - for each g in G that is not consistent with d
            - remove g from G
            - add to G all minimal specialisations h of g such that h is consistent with d and some member of S is more specific than h
            - remove from G any h that is less general than some h’ in G

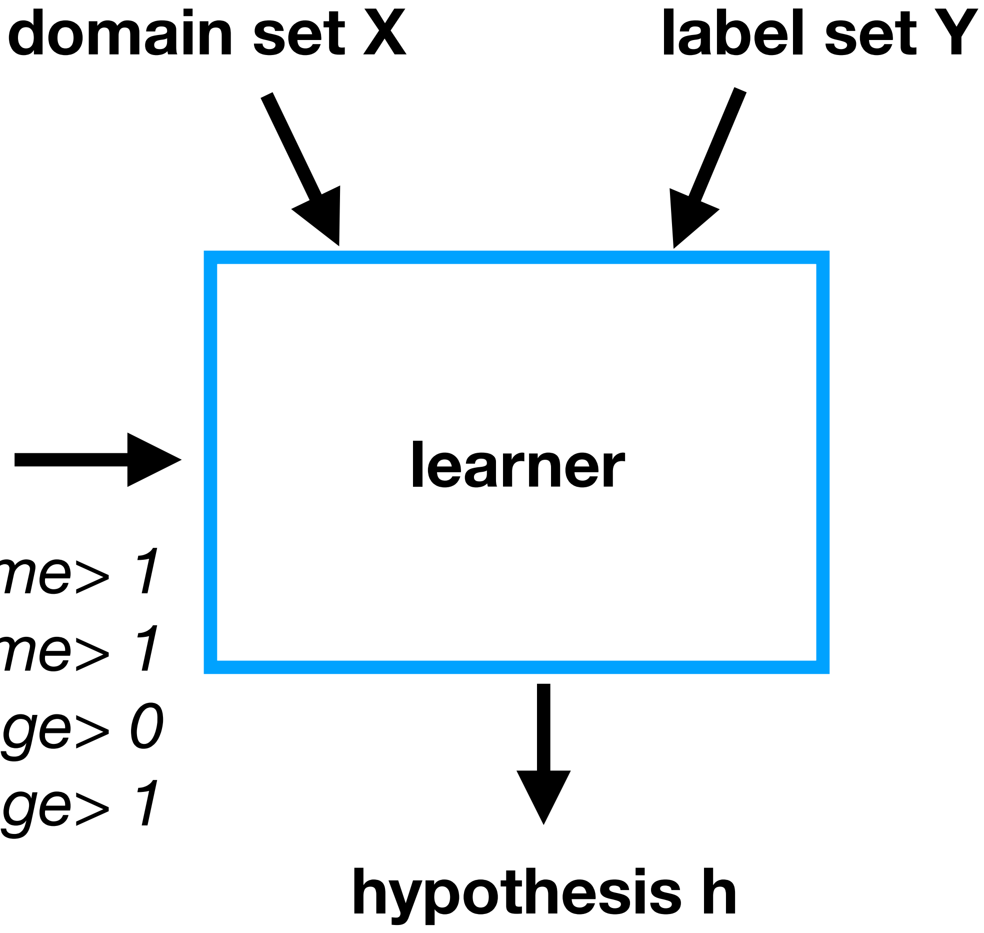
Example

days described by

< Sky, AirTemp, Humidity, Wind, Water, Forecast >



labels 1 / 0 indicating whether someone enjoys their favourite sport on that day



training data D

<Sunny, Warm, Normal, Strong, Warm, Same> 1
<Sunny, Warm, High, Strong, Warm, Same> 1
<Rainy, Cold, High, Strong, Warm, Change> 0
<Sunny, Warm, High, Strong, Cool, Change> 1

conjunction of attribute constraints

G={<?,?,?,?,?,?>}

S={<-,-,-,-,-,->}

<Sunny, Warm, Normal, Strong, Warm, Same> 1
<Sunny, Warm, High, Strong, Warm, Same> 1
<Rainy, Cold, High, Strong, Warm, Change> 0
<Sunny, Warm, High, Strong, Cool, Change> 1

<Sunny, Warm, Normal, Strong, Warm, Same> 1

G={<?,?,?,?,?,?>}

S={<Sunny,Warm,Normal,Strong,Warm,Same>}

<Sunny, Warm, High, Strong, Warm, Same> 1

G={<?,?,?,?,?,?>}

S={<Sunny,Warm,?,Strong,Warm,Same>}

<Rainy, Cold, High, Strong, Warm, Change> 0

G={<Sunny,?,?,?,?,?>, <?,Warm,?,?,?,?>, <?,?,?,?,?,Same>}

S={<Sunny,Warm,?,Strong,Warm,Same>}

<Sunny, Warm, High, Strong, Cool, Change> 1

G={<Sunny,?,?,?,?,?>, <?,Warm,?,?,?,?>}

S={<Sunny,Warm,?,Strong,?,?>}

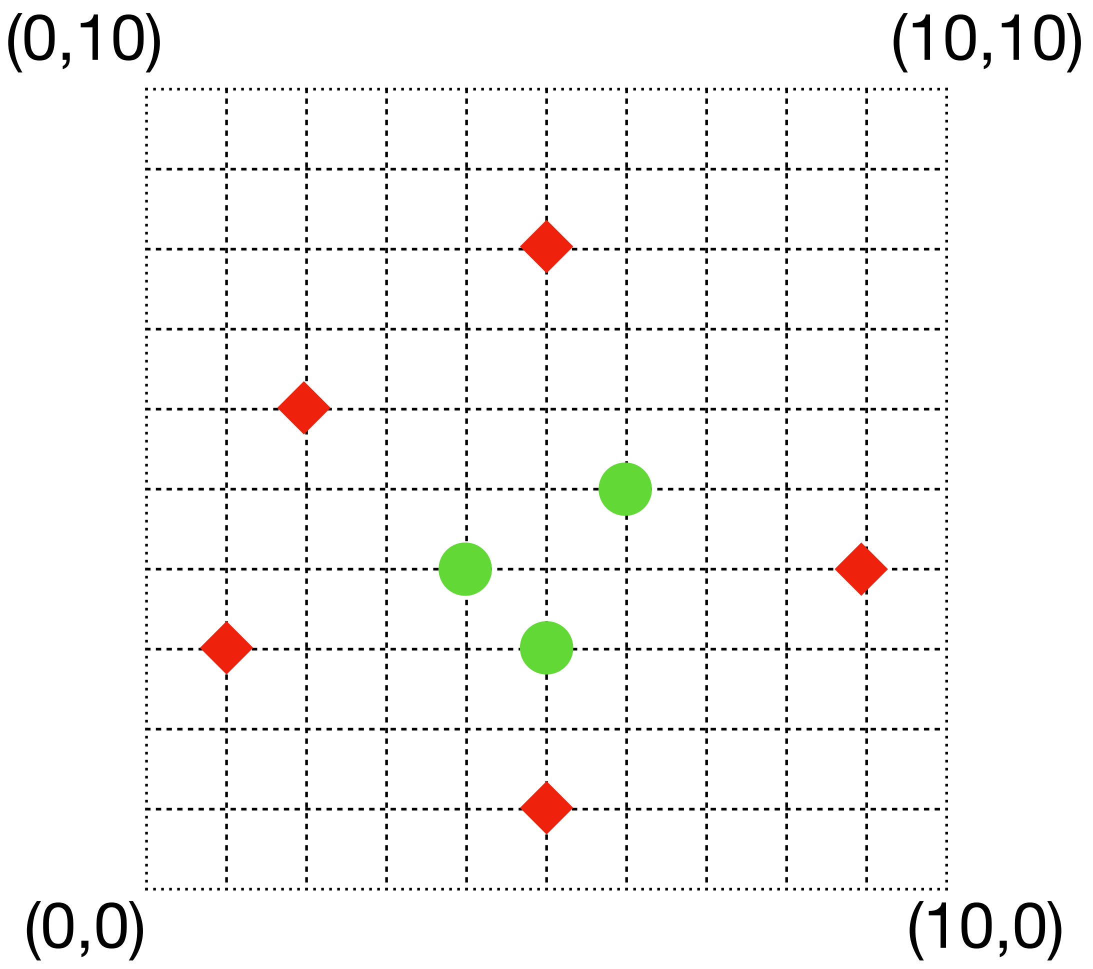
Exercise

- Consider again the space of rectangles ($a \leq x \leq b \wedge c \leq y \leq d$) on the [0,10]x[0,10] grid, and the positive ● and negative ◆ training examples in the figure.

- What are the G and S boundaries of the version space? Write them down and draw them on the grid.

- Imagine the learner can ask the teacher to label a specific point as next training example. Suggest a point that would guarantee to shrink the version space independently of its label, and one that wouldn't.

- What is the smallest number of examples for which CANDIDATE-ELIMINATION can precisely learn any specific rectangle, say, ($2 \leq x \leq 8 \wedge 3 \leq y \leq 5$)?

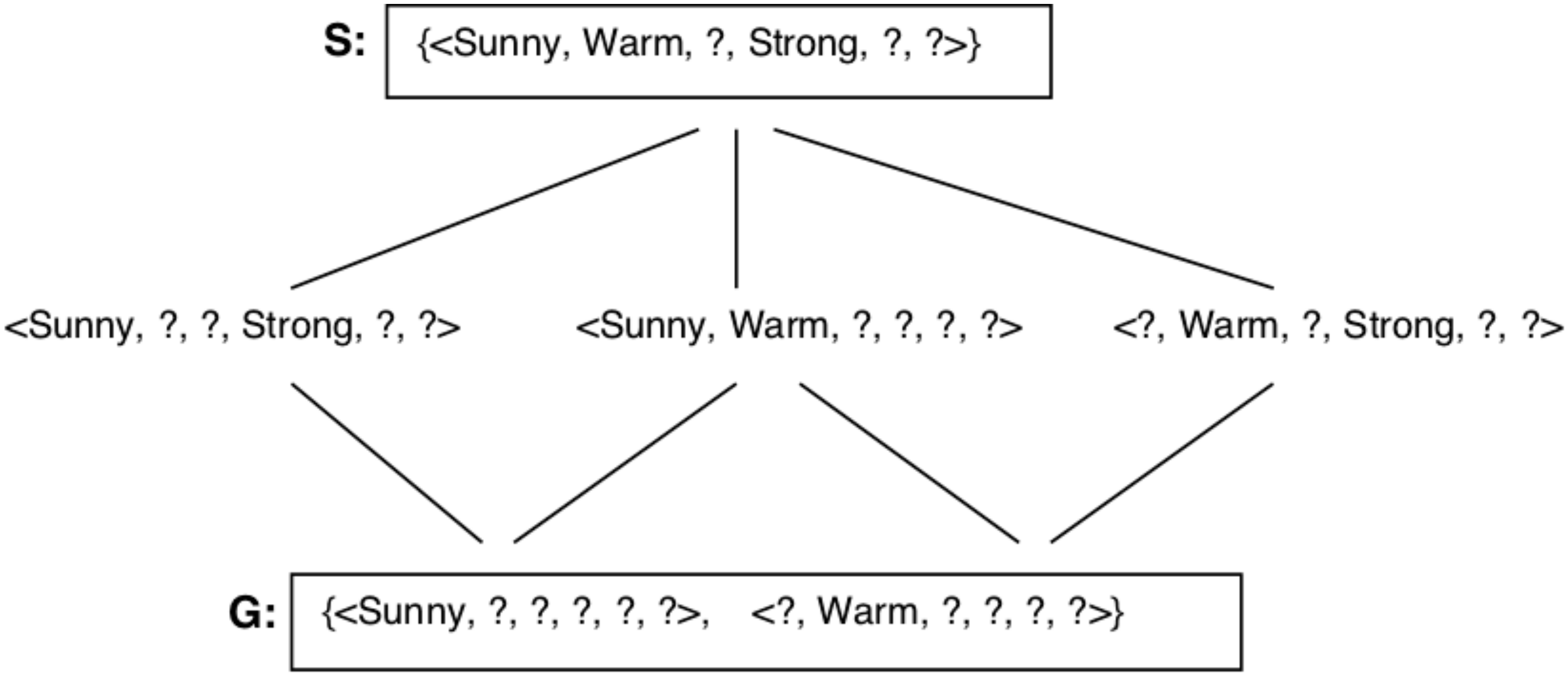


Discussion

- The version space learned by CANDIDATE-ELIMINATION converges towards the hypothesis correctly describing the target concept, provided that
    - there is such a hypothesis in H, and
    - the training data is labeled correctly
- The size of the version space tells us how close we are
- What if we don’t have enough data to converge?
- What if there is no correct h in H?

Using version spaces as classifiers

<Sunny, Warm, Normal, Strong, Cool, Change>
<Rainy, Cold, Normal, Light, Warm, Same>
<Sunny, Warm, Normal, Light, Warm, Same>
<Sunny, Cold, Normal, Strong, Warm, Same>



[Figure: Mitchell]

No correct h in H

days described by

< Sky, AirTemp, Humidity, Wind, Water, Forecast >

Sunny / Cloudy / Rainy

Warm / Cold

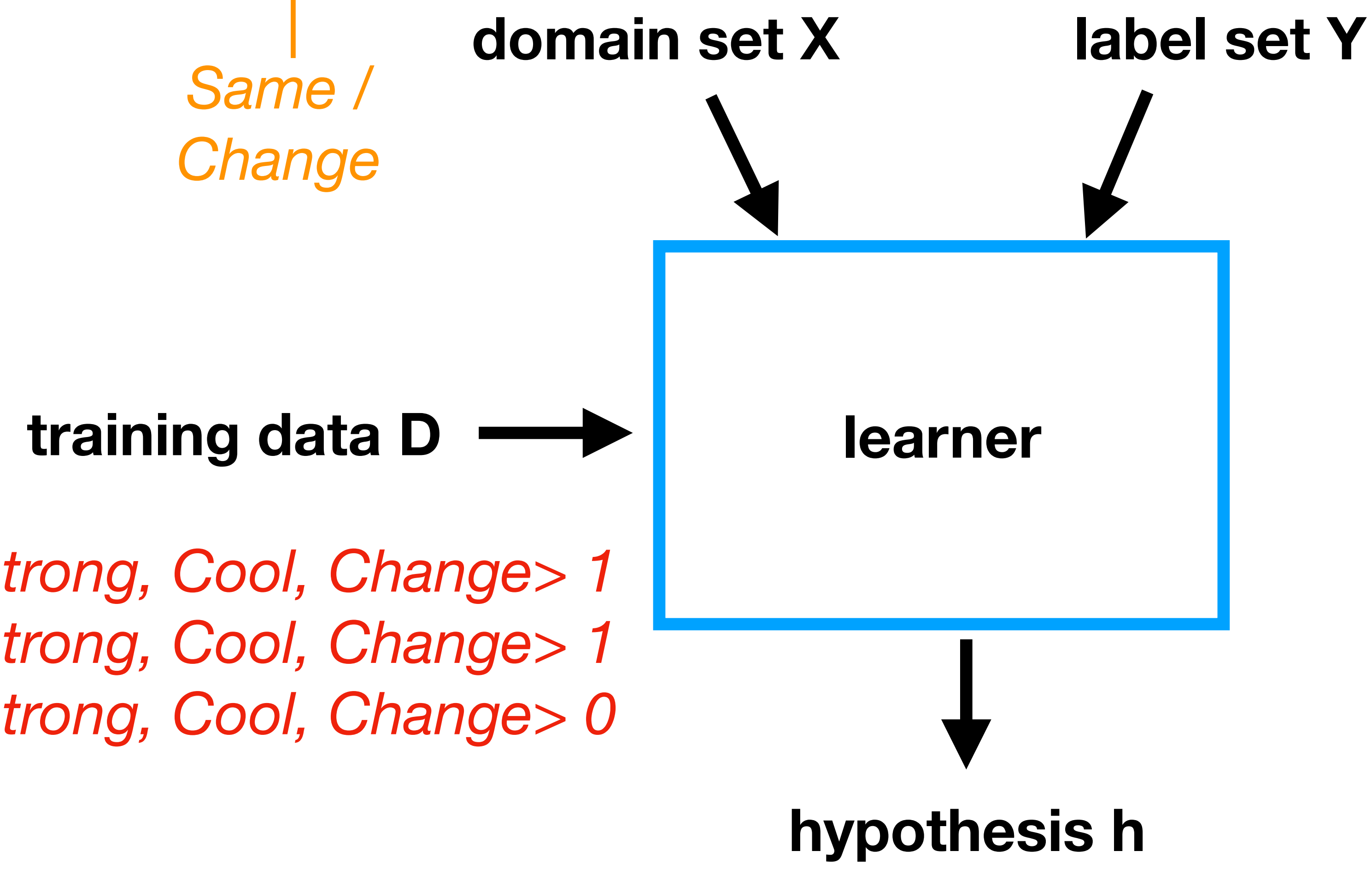
Normal / High

Strong / Weak

Warm / Cool

Same / Change

labels 1 / 0 indicating whether someone enjoys their favourite sport on that day



domain set X

label set Y

training data D

learner

hypothesis h

<Sunny, Warm, Normal, Strong, Cool, Change> 1
<Cloudy, Warm, Normal, Strong, Cool, Change> 1
<Rainy, Warm, Normal, Strong, Cool, Change> 0

conjunction of attribute constraints

No correct h in H

- Problem: there are many more Boolean functions over X than hypotheses in H, so the assumption that there is a good h in H is too strong

- What about including all these functions in H?

- Syntactically, this is easy: just allow any disjunctions, conjunctions and negations of our earlier hypotheses, e.g., <Sunny,?,?,?,?,?> v <Cloudy,?,?,?,?,?>

but…

- CANDIDATE-ELIMINATION now boils down to **memorisation**:
    - S = disjunction of all positive training examples
    - G = negated disjunction of all negative training examples
- only **converges** after **seeing all** instances
- every **unseen** instance is classified **positive by half** of the version space and **negative by the other half**

Inductive bias

- This tension is central to machine learning: we cannot learn **successfully** unless we **restrict** the hypothesis space

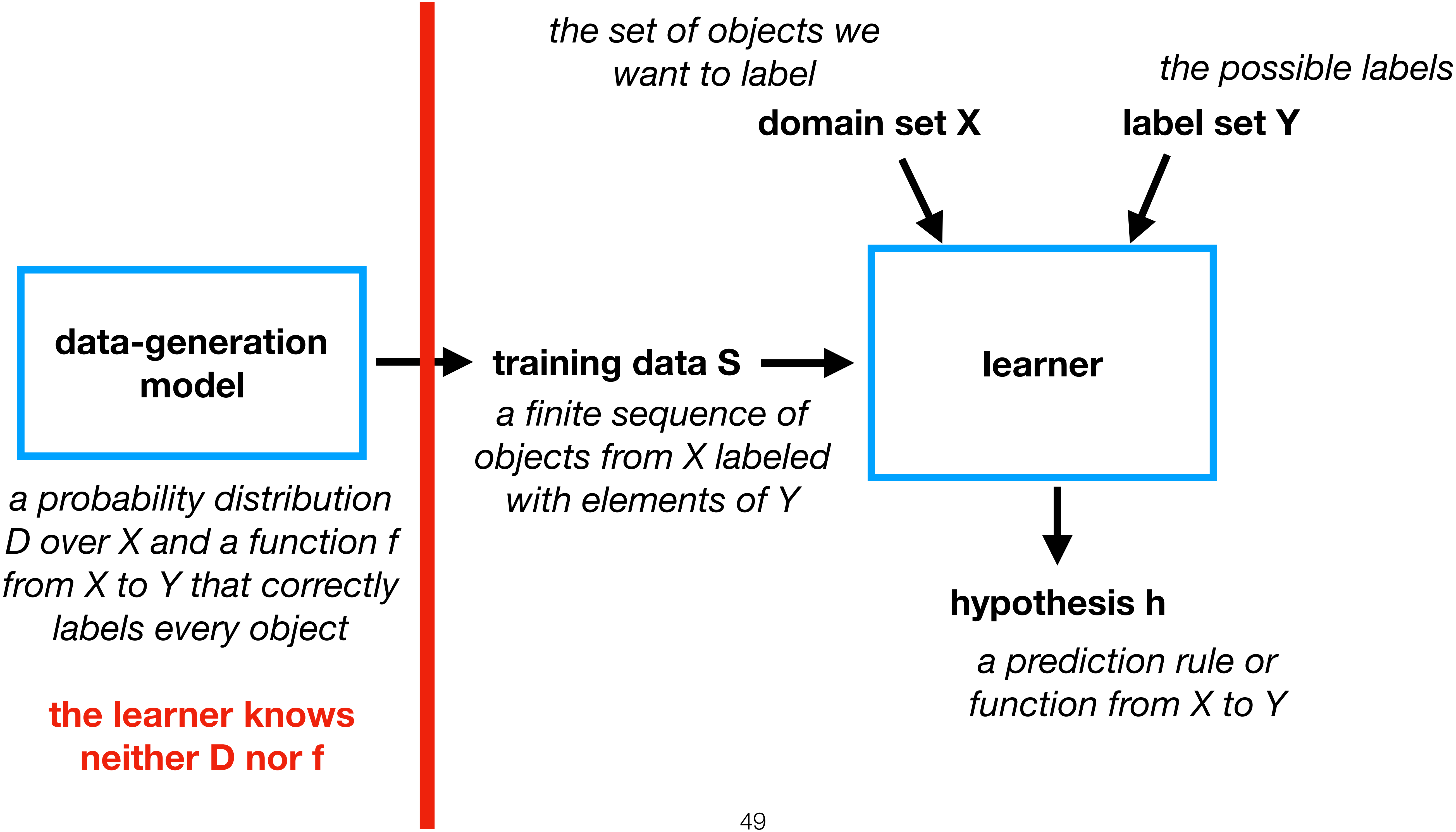
- Different learners make different assumptions to achieve learning; these assumptions are also called **inductive bias**

- Learners with stronger bias make more inductive leaps, classifying larger parts of the instance space

Inductive bias: example

	learning	classification	inductive bias
learner 1	store training data in memory	stored label if available, “unknown” otherwise	none
learner 2	CANDIDATE-ELIMINATION	agreed label if all members of the version space agree, “unknown” otherwise	target concept in hypothesis space
learner 3	FIND-S	label given by learned hypothesis	target concept in H & all examples negative unless there is reason to consider them positive

The Statistical Learning Framework



data-generation model

a probability distribution D over X and a function f from X to Y that correctly labels every object

the learner knows neither D nor f

the set of objects we want to label

domain set X

the possible labels

label set Y

training data S

a finite sequence of objects from X labeled with elements of Y

learner

hypothesis h

a prediction rule or function from X to Y

For next week

- **Mandatory**: revise today’s material
    - relevant textbook chapters:
        - Shalev-Shwartz & Ben-David: chapters 1 & 2.1
        - Mitchell: Chapter 2
- **Optional**: look forward
    - Read Shalev-Shwartz & Ben-David, chapters 2, 3 and 5, with the following questions in mind:
        - What are the key concepts and ideas introduced?
        - How do they relate to the material covered today?