Slide 1

Concept Learning

Lecture Outline:

- What is Concept Learning?
- General-to-Specific Ordering of Hypotheses
- FIND-S: Finding a Maximally Specific Hypothesis

Reading:

Chapter 2 of Mitchell

Slide 2

What is Concept Learning?

- Much learning is acquiring **general** concepts/categories from **specific examples**
  - *house*
  - *game*
  - *driving situations in which I should brake sharply*
  - *credit-worthy loan applicant*
- Such concepts can be viewed as describing
  - a subset over a larger set
    - houses are a particular subset of human artifacts
  - a boolean-valued function over larger set
    - a function over human artifacts which is **true** for houses and **false** for other artifacts
- Definition

  Concept learning: Inferring a boolean-valued function from training examples of its input and output. (Mitchell, p.21)
- Concept learning is also known as **binary classification**

Slide 3

Example: "EnjoySport"

- Suppose want to learn target concept
  days on which Fred enjoys his favorite water sport

- Input is a set of examples, one per day
  - describing the day in terms of a set of *attributes*
  - indicating (yes/no) whether Fred enjoyed his sport that day

Example	Sky	Temp	Humid	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

- Task: learn to predict the value of *EnjoySport* for an arbitrary day, given values of other attributes

Slide 4

Example: "EnjoySport" (cont)

- Suppose hypotheses take form of conjunctions of constraints on instance attributes – e.g. specify allowed values of:
  Sky, Temp, Humid, Wind, Water, Forecast

  and suppose constraints take one of three forms:
  - *?* – any value acceptable
  - `Specified_value` – specific value required, e.g. *Warm* for the *Temp* attribute
  - $\emptyset$ – no value acceptable

- Then, hypotheses can be represented as vectors of such constraints.
  E.g. $\langle ?, Cold, High, ?, ?, ? \rangle$ – represents hypothesis
  Fred enjoys sport only on cold days with high humidity

Slide 5

Concept Learning (cont)

- Can describe the concept learning setting as follows.

  Given:

  – X a set of *instances* over which the concept is to be defined (each represented, e.g., as a vector of attribute values)

  – a *target function* or *concept* to be learned:

  $$c : X \rightarrow \{0, 1\}$$

  – a set D of *training examples*, each of the form $\langle x, c(x) \rangle$ where $x \in X$ and $c(x)$ is the target concept value for x

  (Note: instances in D for which $c(x) = 1$ are called *positive examples*, those for which $c(x) = 0$ are *negative examples*)

  Find:

  – a *hypothesis*, or estimate, of c.

  I.e. supposing H is the set of all hypotheses, find $h \in H$, where $h : X \rightarrow \{0, 1\}$ such that

  $$h(x) = c(x) \text{ for all } x \in X$$

- Thus, concept learning can be viewed as **search** over the space of hypotheses, as defined by the hypothesis representation.

Slide 6

Concept Learning (cont)

- Note: we want to find h identical to c over all of X **but**, only have information about c for training examples D.

- Inductive learning algorithms can only guarantee that hypotheses fit *training data*.

- Proceed under following assumption (Mitchell, p. 23):

  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.

Slide 7

General-to-Specific Ordering of Hypotheses

- Recall concept learning may be viewed as search over the space of hypotheses, as defined by the hypothesis representation.

- For the *EnjoySport* example there are
  - 6 attributes (*Sky, Temp, Humid, Wind, Water, Forecast*)
  - with $3 \times 2 \times 2 \times 2 \times 2 \times 2 = 96$ distinct value combinations

- Since the hypothesis representation allows each attribute to take ? or $\emptyset$ in addition to specific values, there are: $5 \times 4 \times 4 \times 4 \times 4 \times 4 = 5120$ *syntactically distinct* hypotheses.

- However, since every hypothesis containing one or more $\emptyset$'s is equivalent (classifies all instances as negative) this reduces to $1 + 4 \times 3 \times 3 \times 3 \times 3 \times 3 = 973$ *semantically distinct* hypotheses.

Slide 8

General-to-Specific Ordering of Hypotheses (cont)

- **Problem**: How should we search this space to find the target concept?

  A Solution: Start with the most specific hypothesis and, considering each training example in turn, generalise towards the most general hypothesis, stopping at the first hypothesis that 'covers' the training examples (FIND-S)
  - the most specific hypothesis:

    $$\langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle \text{ (\textit{Fred enjoys sport on no day})}$$

  - the most general hypothesis:

    $$\langle ?, ?, ?, ?, ?, ? \rangle \text{ (\textit{Fred enjoys sport on every day})}$$

Slide 9

General-to-Specific Ordering of Hypotheses (cont)

- Intuitively a hypothesis h_i is more general than another h_j if
  - every instance that h_j classifies as positive h_i also classifies as positive, and
  - h_i classifies instances as positive that h_j does not

  E.g.

$$\begin{aligned} h_1 &= \langle Sunny, ?, ?, Strong, ?, ? \rangle \\ h_2 &= \langle Sunny, ?, ?, ?, ?, ? \rangle \end{aligned}$$

  h_2 is more general than h_1

- More formally, instance x **satisfies** hypothesis h iff $h(x) = 1$.

  Define partial ordering relation *more_general_than_or_equal_to* holding between two hypotheses h_i and h_j in terms of the sets of instances that satisfy them:

  h_i is *more_general_than_or_equal_to* h_j iff every instance that satisfies h_j also satisfies h_i

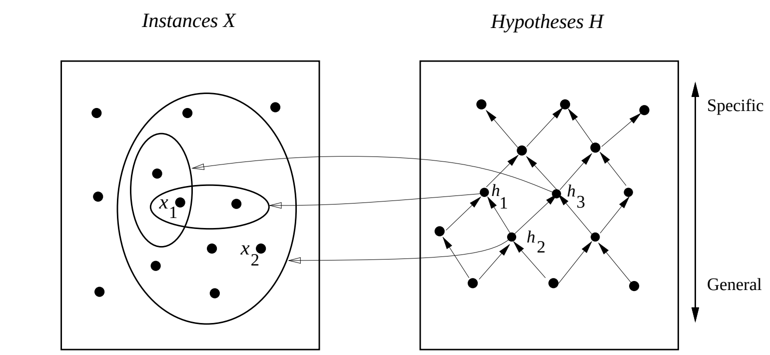
  or

  Definition: Let h_i and h_j be boolean-valued functions defined over X. h_i is **more_general_than_or_equal_to** h_j (written $h_i \geq_g h_j$) if and only if

$$(\forall x \in X)[(h_j(x) = 1) \rightarrow (h_i(x) = 1)]$$

Slide 10

General-to-Specific Ordering of Hypotheses (cont)



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, ?>

- Note that in the example:

$$\begin{aligned} h_2 &\geq_g h_1 \qquad & h_2 &\geq_g h_3 \\ h_1 &\not\geq_g h_3 \qquad & h_3 &\not\geq_g h_1 \end{aligned}$$

- $\geq_g$ is useful because it provides a structure over hypothesis space H for any concept learning problem. Learning algorithms can take advantage of this …

Slide 11

FIND-S: Finding a Maximally Specific Hypothesis

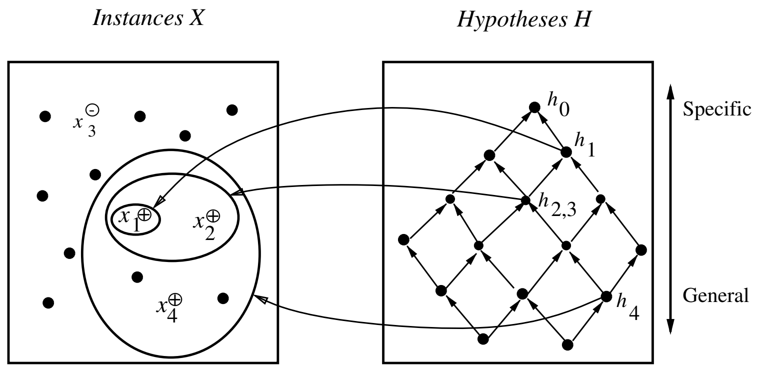
- FIND-S algorithm:

> 1. Initialize h to the most specific hypothesis in H
> 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

Slide 12

FIND-S: Finding a Maximally Specific Hypothesis (cont)

- For *EnjoySport* training examples:



h_0 = <∅, ∅, ∅, ∅, ∅, ∅>

x_1 = <*Sunny Warm Normal Strong Warm Same*>, + — h_1 = <*Sunny Warm Normal Strong Warm Same*>

x_2 = <*Sunny Warm High Strong Warm Same*>, + — h_2 = <*Sunny Warm ? Strong Warm Same*>

x_3 = <*Rainy Cold High Strong Warm Change*>, - — h_3 = <*Sunny Warm ? Strong Warm Same*>

x_4 = <*Sunny Warm High Strong Cool Change*>, + — h_4 = <*Sunny Warm ? Strong ? ?* >

- Note: negative training instances completely ignored by FIND-S

Slide 13

Find-S: Finding a Maximally Specific Hypothesis (cont)

- For hypothesis spaces described by conjunctions of attribute constraints (e.g. H for *EnjoySport*), FIND-S is guaranteed to output the most specific hypothesis in H consistent with positive training examples
- FIND-S also guaranteed to output hypothesis consistent with negative examples provided
  - correct target concept is in H
  - training examples are correct
- But, there are problems with FIND-S:
  - may be multiple hypotheses consistent with the training data – FIND-S will find one, but give no indication of whether there may be others
  - FIND-S always proposes maximally specific hypothesis – why prefer this to, e.g., maximally general?
  - FIND-S has serious problems when training examples are inconsistent which frequently happens with noisy "real" data

Slide 14

Summary

- A basic, but important, class of machine learning problems may be described as **concept learning** – learning a general concept from specific examples
- Concept learning may also be viewed as inferring a boolean-valued function from training examples.

  The general setting is, given an instance space X
  - a target function to be learned: $c : X \rightarrow \{0,1\}$
  - a set D of training examples to learn from, each of the form $\langle x, c(x) \rangle$, where $x \in X$
  - a hypothesis $h : X \rightarrow \{0,1\}$ which is an approximation of c – ideally $h(x) = c(x)$ for all $x \in X$
- Hypotheses h are drawn from some hypothesis space H. Much of machine learning involves exploring techniques for searching H

Slide 15

Summary (cont)

- One sort of technique revolves around realising that hypotheses can be **ordered from general-to-specific**

  The intuition here is that one hypothesis is more general than another if every instance the more specific hypothesis classifies as positive the more general one also classifies as positive, as well as classifying others as positive too

- Algorithms to search the hypothesis space can make use of this ordering

  For example **FIND-S** searches from the most specific to the most general hypothesis, stopping when it finds the most specific hypothesis that covers all of the training examples.