Chapter 2 Concept Learning

- What does Concept Learning mean?
- How to perform Concept Learning?
- Some remarks?

What does Concept Learning mean?

Definition: Inferring a Boolean-valued function from the given examples of inputs and outputs. E. g.,

- If the module is interesting, then I will like it.
- If the module involves heavy maths, lack of notes, and too muck work, then I do not like this module.

The Training Examples: Attitude toward IL Module

No	Math	Motivation	Presentation	Applications	References	EnjoyIL
1	Strong	Positive	Interesting	Wide	Rich	Yes
2	Weak	Positive	Interesting	Wide	Rich	Yes
3	Strong	Negative	Interesting	Narrow	Rich	Yes
4	Heavy	Negative	Dull	Narrow	Poor	no

What is the target concept?

Type of Examples

- Positive examples: c(x)=1
- Negative examples: c(x)=0

E.g., 1=LikeIL; 0=DislikeIL. The examples that have the conclusion saying that students like IL are positive examples. The examples that have the conclusion saying that students do not like IL are negative examples.

Some useful symbols

Consider five constraints: Math, Motivation, Presentation, Application, and Reference:

- $?$ indicates that any value for the attribute is acceptable.
- $\emptyset$ indicates that no value for the attribute is acceptable.
- $<?, ?, ?, ?, ?>$ denotes that whatever happens, I like the Intelligent Learning Module.
- $<\emptyset, \emptyset, \emptyset, \emptyset, \emptyset>$ denotes that whatever happens, I do not like Intelligent Learning Module.

Representation hypotheses

Here, hypothesis h is a conjunction of constraints on attributes
Each attribute can be:

- a specific value (e.g., Motivation=Positive)
- don't care (e.g., Motivation=?)
- no value is allowed (e.g., Motivation=$\emptyset$)

For example,

Math	Motivation	Presentation	Applications	References	EnjoyIL
Heavy	Negative	Dull	?	?	No

Representing Concept Learning Problem

- **Given** :
  - Instances X: Possible conditions, each described by the attributes
    - Math (with possible values: Strong, Weak, Heavy)
    - Motivation (with possible values: Positive, Negative)
    - Presentation (with possible values: Interesting, Dull)
    - Applications (with possible values: wide, Narrow)
    - References (with possible values: Rich, Poor)

- Hypotheses H: each hypothesis is described by a conjunction of constraints on the attributes Math, Motivation, Presentation, Applications, and References. The constraints could be $?$(any value is acceptable), $\emptyset$(no value is acceptable), or a specific value.
- Target Function c: EnjoyIL: $X \rightarrow \{0, 1\}$
- 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 X.
- Examples
  - If this module involves too much work, I would probably start to dislike this module.
  - If the module is interesting, well-presented, not too difficult, and applicable, then I like this module.
  - If we can do something practical with what we have learnt, I will like it.
  - If the module is easy to understand, then I like it.

Concept Learning as a Search

For the conditions under which you like or dislike Intelligent Learning module, There are five attributes: Math, Motivation, Presentation, Applications, and References.

- Math has three possible values: Strong, Weak, and Heavy.
- Motivation has two possible values: Positive, Negative
- Presentation has two possible values: Interesting, Dull
- Applications has two possible values: Wide, Narrow
- References has two values: Rich, Poor

- Since any combination of these attribute values can make up a legal hypothesis, then we have: 3*2*2*2*2=48.

$$H = \{H_1, H_2, \cdots, H_{48}\}$$

- If we consider that any combination of these hypotheses could be the final learned hypotheses, than we have to find the power of H which has $2^{|H|} - 1 = 2^{48} - 1$ elements.
- Actually, the hypothesis space we represented has just: 1+4*3*3*3*3=325 hypotheses.

- Definition: x satisfies h if and only if h(x)=1.
- Definition: Let h_i and h_j be two Boolean-valued functions defined over X. Then h_j is $more-general-than-or-equal-to$ h_i(written as $h_j \geq h_i$) if and only if $(\forall x \in X)[(h_i(x) = 1) \rightarrow (h_j(x) = 1)]$
- For exemple,
  - h_i=<Strong, Positive, Interesting, Wide, Rich>
    h_j=<?, Positive, Interesting, Wide, Rich>
  - h_i=<I am good at learning IL>
    h_j=<I am good at learning>

  In these two cases: h_i can be generalised to h_j and h_j can be specialised to h_i

The Inductive Learning Hypothesis

- **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.

Two Operators for Learning

Whenever getting stuck, considering the following:

- Generalisation:
  - Remove some constraints
  - Introduce new variables
  - Introduce disjunctions
- Specialisation:
  - Add some constraints
  - Instantiate variables with specific values
  - Introduce conjunctions

Rote Learning

- Learn by heart without thinking of the meaning
- Look up tables

FIND-S algorithm

- Initialize h to the most specific hypothesis in H
- For each positive training instance x

  For each attribute value 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
- Output hypothesis h

An Example: Attitude toward IL Module

- In the four rules, they are equally specific, so we can randomly select any one as h_0, for example,

  h_0=<Strong, Positive, Interesting, Wide, Rich>

- Then we use the second rule

  <Weak, Positive, Interesting, Wide, Rich>
  for learning: In order to make h_0 cover the second rule, we have to search the hypothesis space H and find the one which is as specific as possible but is able to cover the second rule. So we have

  h_1=<?, Positive, Interesting, Wide, Rich>

- Then we use the third rule

  <Strong, Negative, Interesting, Narrow, Rich>
  for learning: In order to make h_1 cover the third rule, we have to search the hypothesis space H again and find the one which is as specific as possible but is able to cover the third rule. So we have

  h_2=<?, ?, Interesting, ?, Rich>
- The fourth rule is a negative one. We could directly throw it.

- Output the finally learned rule:

  h=<?, ?, Interesting, ?, Rich>

- Problem: based on the learned rule, can we predict whether a student like the Intelligent Learning module under the conditions:

  <Weak, Negative, Interesting, Narrow, Rich>

Some Points about FIND-S Algorithm

- FIND-S works only when:
  - H contains a hypothesis that describes the true target concept
  - No errors in training data
  - Hypotheses in H described as conjunctions of attribute constraints
- Complaints:
  - Can't tell whether the algorithm has converged to the correct solution
  - Can't detect whether training data are inconsistent
  - Picks a maximally specific h (why?)
  - Depending on H, there might be several!

CANDIDATE ELIMINATION Algorithm

- This algorithm finds all describable hypotheses that are consistent with the observable examples.
- Definition: A hypothesis h is consistent with a set of training examples if and only if $h(x) = c(x)$ for each example $< x, c(x) >$ in D.

$$Cosistent(h, D) \equiv (\forall < x, c(x) > \in D) h(x) = c(x)$$

Version Space

Definition: The version space, denoted $VS_{H,D}$, with respect to hypothesis Space H and training examples D, is the subset of hypotheses from H consistent with the training examples D.

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

The LIST-THEN-ELIMINATE Algorithm

1. VersionSpace $\leftarrow$ a list containing every hypothesis in H
2. For each training example, $< x, c(x) >$,
   - Remove from VersionSpace any hypothesis h for which $h(x) \neq c(x)$
3. Output the list of hypotheses in VersionSpace

Comments of LIST-THEN-ELIMINATE Algorithm

- **Strong point**: It is guaranteed to output all hypotheses consistent with the training examples.
- **Weak point**: It requires explicitly enumerating all hypotheses in H.

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:

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

Analysis of Inductive Bias

- Search space(Representational bias): Is the target concept in the hypothesis space H?
  - Whether the method chosen to represent hypotheses are expressive enough to represent all possible hypotheses?
- Search strategy(Search bias): How to search the desired hypotheses in the search space: from most specific ones to most general ones? Or vice versa.
  - Prefer hypotheses with most concise expression?
  - Prefer first discovered consistent hypotheses?

Three Learners with Different Biases

- *Rote learner:* Store examples, Classify x iff it matches previously observed example. Otherwise, the algorithm refuses to classify the new instances.
- *Version space candidate elimination algorithm*: New instances are classified only when the all member of the current version space agree on the classification. Otherwise, the system refuses to classify the new instance.
- *Find-S*: Use the most specific hypothesis to classify all subsequent instances. If the algorithm cannot classify them as positive, then the algorithms assumes that the new instances are negative.

Summary Points

- Concept learning can be cast as search through H.
- General-to-specific ordering can be used to organise the search over H.
- FIND-S algorithm uses general-to-specific ordering to find the most specific hypothesis consistent with the training examples.
- List-Then-Eliminate can output a number of hypotheses consistent with the training examples. But it requires explicitly enumerating all possible hypotheses.
- Learner can generate useful queries.

Questions

- What does Concept Learning mean?
- State three algorithms for concept learning.
- What is inductive bias? How to analyse it?
- How to analyse the inductive bias of rote, FIND-S, and List-Then-Eliminate algorithms?