Chap. 19

Knowledge in Learning

In which we examine the problem of learning
when you know something already (i.e. prior knowledge).

1

Outline

- Pure inductive learning:
  a process of finding a consistent hypothesis in the hypothesis space.
- Logical Formulation of Learning
  - The hypothesis is represented by a set of logical sentences:
    Classification of new example by *inference* of classification sentence
    from hypothesis and the example description.
  - It allows for Incremental construction of hypothesis and Prior knowledge.
- Current Best Hypothesis Search
  - Maintain a single hypothesis, and to adjust it as new examples arrive in order to maintain consistency.
- Version Space Search (Least-Commitment Search)

2

Learning
General Logical Descriptions

- Process of searching for a good hypothesis in *hypothesis space* defined by the representation language.

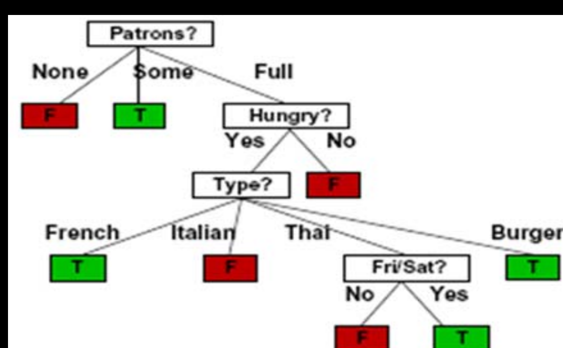
- $Q(x)$: Goal predicate (classification: $Q(x)$, $\neg Q(x)$)
- $C_i(x)$: Candidate definition $\forall x\ Q(x) \Leftrightarrow C_i(x)$
- H_i: Hypothesis in the form

- $D_i(x_i)$: description of an example x_i.

- Aim:
  To find an equivalent logical expression for Q that we can use to classify examples correctly – a *candidate definition of Q*.

3

Example



$\forall r\ WillWait(x) \Leftrightarrow Patrons(r, Some)$
$\quad \vee (\ Patrons(r, Full) \wedge \neg Hungry(r) \wedge Type(r, French)\ )$
$\quad \vee (\ Patrons(r, Full) \wedge \neg Hungry(r) \wedge Type(r, Thai) \wedge Tri/Sat(r)\ )$
$\quad \vee (\ Patrons(r, Full) \wedge \neg Hungry(r) \wedge Type(r, Burger)\ )$

4

Hypothesis Space

- The *extension* of the predicate: each hypothesis predicts that a certain set of examples are those of goal predicate.
  - 2 hypotheses with different extensions are inconsistent: disagreement on their predictions for at least 1 example.
- H: Hypothesis Space = $\{h_1, h_2, \ldots, h_n\}$
  - The set of all hypotheses that the learning algorithm is designed to entertain.
- Learning Algorithm believes that the sentence

$$h_1 \lor h_2 \lor h_3 \lor \ldots \lor h_n$$

  is correct.
- As the examples arrives, rule out inconsistent hypotheses.

5

Example

- The classification $Q(X_i)$ if X_i is a positive example
  $\neg Q(X_i)$ if X_i is a negative example

Example	Attributes										Goal
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
X1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	Yes
X2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	No
X3	No	Yes	No	No	Some	$	No	No	Burger	0-10	Yes
X4	Yes	No	Yes	Yes	Full	$	No	No	Thai	10-30	Yes
X5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	No
X6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	Yes
X7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	No
X8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	Yes
X9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	No
X10	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	No
X11	No	No	No	No	None	$	No	No	Thai	0-10	No
X12	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	Yes

The first line $(D_1(X_1))$ — classification

$Alternate(Xi) \land \neg Bar(Xi) \land Tri/Sat(Xi) \land Hungry(Xi) \land \ldots$ — $WillWait(Xi)$

6

Consistency

- A hypothesis agrees with all the examples if and only if it is logically consistent with the training set.
- Two types of inconsistency :
  - False negative : if the hypothesis says it should be negative but in fact it is positive.

  e.g.) X_{13}: $Patrons(X_{13}, Full) \wedge \mathrm{Wait}(X_{13}, 1\text{-}10) \wedge \neg Hungry(X_{13}) \wedge WillWait(X_{13})$

  - False positive : if the hypothesis says it should be positive but in fact it is negative.
- If an example is a false positive or false negative for a hypothesis, then hypothesis and the example are inconsistent.
- Analogous to Resolution of inference: If an example is inconsistent with h_i, the inference system can deduce the new h removing h_i.
- Inductive learning can be characterized in a logical setting as a process of gradually eliminating inconsistent hypotheses with the examples, narrowing down the possibilities.

7

Current-Best-Hypothesis Search

1. Maintain a single hypothesis.
2. Adjust the hypothesis as new examples arrive.

- Generalization:

  If a false negative example arrived, the hypothesis says it should be negative, but it is actually positive.

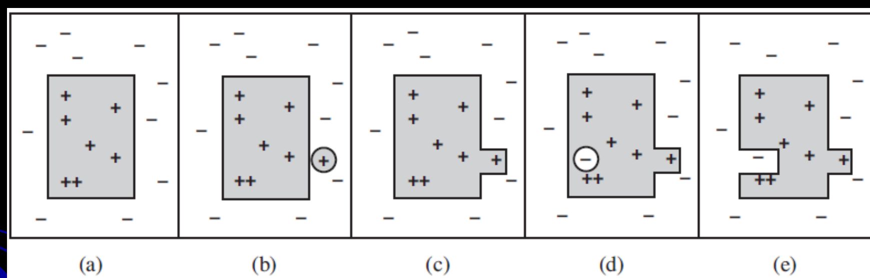
  The extension of the hypothesis must be increased.
- Specialization:

  If a false positive example arrived, the hypothesis says it should be positive, but it actually negative.

  The extension of the hypothesis must be decreased.

8

Possible Extensions



(a) A consistent hypothesis
(b) A false negative
(c) The hypothesis is generalized
(d) A false positive
(e) The hypothesis is specialized

Current-Best-Hypothesis Search Algorithm

```
function CURRENT-BEST-LEARNING(examples, h) returns a hypothesis or fail

  if examples is empty then
     return h
  e ← FIRST(examples)
  if e is consistent with h then
     return CURRENT-BEST-LEARNING(REST(examples), h)
  else if e is a false positive for h then
     for each h′ in specializations of h consistent with examples seen so far do
        h″ ← CURRENT-BEST-LEARNING(REST(examples), h′)
        if h″ ≠ fail then return h″
  else if e is a false negative for h then
     for each h′ in generalizations of h consistent with examples seen so far do
        h″ ← CURRENT-BEST-LEARNING(REST(examples), h′)
        if h″ ≠ fail then return h″
  return fail
```

- CBH search algorithm searches for a consistent hypothesis and backtracks when no consistent specialization/generalization can be found.
- Specialization/Generalization:
  operations that change the extension of a hypothesis.

```
function CURRENT-BEST-LEARNING(examples, h) returns a hypothesis or fail

  if examples is empty then
    return h
  e ← FIRST(examples)
  if e is consistent with h then
    return CURRENT-BEST-LEARNING(REST(examples), h)
  else if e is a false positive for h then
    for each h' in specializations of h consistent with examples seen so far do
      h'' ← CURRENT-BEST-LEARNING(REST(examples), h')
      if h'' ≠ fail then return h''
  else if e is a false negative for h then
    for each h' in generalizations of h consistent with examples seen so far do
      h'' ← CURRENT-BEST-LEARNING(REST(examples), h')
      if h'' ≠ fail then return h''
  return fail
```

Dropping condition

- How to change the candidate definition associated with the hypothesis using generalization/specialization? - a logical relationships b/t hypotheses.
- Let's assume:

  Hypothesis h_1, with definition C_1, is a generalization of
  Hypothesis h_2 with definition C_2, then

  $$\forall x\ C_2(x) \Rightarrow C_1(x)$$

- Example : (Dropping Condition)

  One possible generalization for C_2 is C_1.

  $C_2(x) \equiv Alternate(x) \wedge Patrons(x, Some)$ $C_1(x) \equiv Patrons(x, Some)$ - weaker.

- In order to construct a generalization of h_2, find a definition C_1 that is logically implied by C_2: Dropping conditions.

12

Current-Best-Hypothesis Search Example

Example	Attributes										Goal
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
X1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	Yes
X2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	No
X3	No	Yes	No	No	Some	$	No	No	Burger	0-10	Yes
X4	Yes	No	Yes	Yes	Full	$	No	No	Thai	10-30	Yes
X5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	No
X6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	Yes
X7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	No
X8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	Yes
X9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	No
X10	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	No
X11	No	No	No	No	None	$	No	No	Thai	0-10	No
X12	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	Yes

- The first example is positive.
  $Alternate(X_1)$ is true, so initial hypothesis can be,

  h_1: $\forall x\ WillWait(x) \Leftrightarrow Alternate(x)$

13

Example: continued..

Example	Attributes										Goal
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
X1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	Yes
X2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	No
X3	No	Yes	No	No	Some	$	No	No	Burger	0-10	Yes
X4	Yes	No	Yes	Yes	Full	$	No	No	Thai	10-30	Yes
X5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	No
X6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	Yes
X7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	No
X8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	Yes
X9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	No
X10	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	No
X11	No	No	No	No	None	$	No	No	Thai	0-10	No
X12	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	Yes

- The second example is negative. H_1 predicts it to be positive, so false positive.
  We need to *specialize* h_1. One possible specialization is:

  h_2: $\forall x\ WillWait(x) \Leftrightarrow Alternate(x) \wedge Patrons(x, Some)$

14

Example: continued..

Example	Attributes										Goal
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
X1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	Yes
X2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	No
X3	No	Yes	No	No	Some	$	No	No	Burger	0-10	Yes
X4	Yes	No	Yes	Yes	Full	$	No	No	Thai	10-30	Yes
X5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	No
X6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	Yes
X7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	No
X8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	Yes
X9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	No
X10	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	No
X11	No	No	No	No	None	$	No	No	Thai	0-10	No
X12	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	Yes

- The 3rd example is positive. H_2 predicts it to be negative, so false negative. We need to *generalize* h_2. One possible generalization is:

  h_3: $\forall x$ WillWait(x) $\Leftrightarrow$ Patrons(x, Some)

Example: continued..

Example	Attributes										Goal
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
X1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	Yes
X2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	No
X3	No	Yes	No	No	Some	$	No	No	Burger	0-10	Yes
X4	Yes	No	Yes	Yes	Full	$	No	No	Thai	10-30	Yes
X5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	No
X6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	Yes
X7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	No
X8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	Yes
X9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	No
X10	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	No
X11	No	No	No	No	None	$	No	No	Thai	0-10	No
X12	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	Yes

- The 4th example is positive. H_3 predicts it to be negative, so false negative. We need to generalize h_3. We can not drop Patrons condition, because that would yield an inclusive hypothesis inconsistent with 2nd example, one possibility

  h_4: $\forall x$ WillWait(x) $\Leftrightarrow$ Patrons(x, Some) $\vee$ (Patrons(x, Full) $\wedge$ Fri/Sat(x))

Example: continued..

Example	Attributes										Goal
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
X1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0-10	Yes
X2	Yes	No	No	Yes	Full	$	No	No	Thai	30-60	No
X3	No	Yes	No	No	Some	$	No	No	Burger	0-10	Yes
X4	Yes	No	Yes	Yes	Full	$	No	No	Thai	10-30	Yes
X5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	No
X6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0-10	Yes
X7	No	Yes	No	No	None	$	Yes	No	Burger	0-10	No
X8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0-10	Yes
X9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	No
X10	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10-30	No
X11	No	No	No	No	None	$	No	No	Thai	0-10	No
X12	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30-60	Yes

- h_4: $\forall x\ WillWait(x) \Leftrightarrow Patrons(x, Some) \vee (Patrons(x, Full) \wedge Fri/Sat(x))$
- Obviously, there are other possibilities consistent with $X_1 - X_4$.:

  $h_{4'}$: $\forall x\ WillWait(x) \Leftrightarrow \neg WaitEstimate(x, 30\text{-}60)$

  $h_{4''}$: $\forall x\ WillWait(x)$

  $\Leftrightarrow Patrons(x, Some) \vee (Patrons(x, Full) \wedge WaitEstimate(x, 10\text{-}30))$

17

Drawbacks of Current-Best-Hypothesis Search Algorithm

1. Nondeterministic
2. Need Backtracking

- At any point there may be several possible extensions.
- In some cases, it is hard, sometimes impossible, to find consistent hypothesis with all of the data.

- Difficulty:

  1. Checking all the previous examples over again for each modification is very expensive.

  2. The search process may involve a great deal of backtracking.

18

Least-commitment Search

- Assume the original hypothesis space contains the consistent hypotheses so far. H = $\vee_i h_i$.
- Each new instance will either have no effect or will remove some h_is.
- Reduced disjunction must still contain the consistent hypotheses because only incorrect hypotheses have been removed.
- The set of hypothesis remaining is called *version space*.
- The learning algorithm is called
  version space learning algorithm
  or *candidate elimination algorithm.*

19

Version-Space-Learning Algorithm

function VERSION-SPACE-LEARNING(*examples*) **returns** a version space
  local variables: V, the version space: the set of all hypotheses

  $V \leftarrow$ the set of all hypotheses
  for each example e in *examples* **do**
    if V is not empty **then** $V \leftarrow$ VERSION-SPACE-UPDATE(V, e)
  return V

---

function VERSION-SPACE-UPDATE(V, e) **returns** an updated version space
  $V \leftarrow \{h \in V : h \text{ is consistent with } e\}$

- It finds a subset V that is consistent with the examples.

20

Version-Space-Learning Algorithm

- Properties:
  - *Incremental*

    One never has to go back and reexamine the old examples.
  - *Least-Commitment*

    It makes no arbitrary choices.

    Problem : Hypothesis space is enormous, how can we possibly write down this enormous disjunction?

    Solution : Use real numbers analogy

    -- Use intervals and have an ordering on the values.

    : an *ordering* on the hypothesis space

21

Boundary Set

- We also have an ordering on the hypothesis space, namely, *generalization/specialization*.
- This is a *partial ordering*:

  Each boundary will not be a point but rather a set of hypotheses, boundary set.
- We can represent the entire version space by two boundary sets,

  G-Set : The most general set.

  S-Set : The most specific set.

  Everything b/t G-set and S-set is guaranteed to be consistent with the examples.

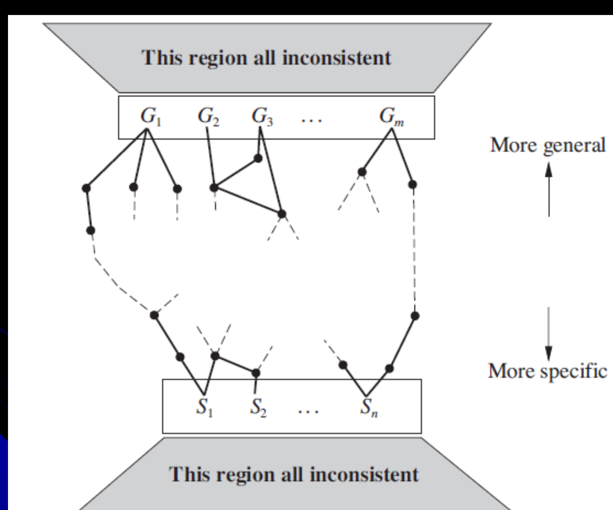
22

G-Set and S-Set

- The current version space is the set of hypotheses consistent with all the examples so far, represented by S-Set and G-Set.
- Every member of S-Set is consistent with all observations so far, and there are no consistent hypotheses that are more specific.
- Every member of G-Set is consistent with all observations so far, and there are no consistent hypotheses that are more general.
- Properties:
  - Every consistent hypothesis other than those in G/S-sets is more specific than some $G_j s \in$ G-set and more general than $S_k s \in$ S-set.
  - Every hypothesis more specific than $G_j s \in$ G-sets and more general than $S_j s \in$ S-set is a consistent hypothesis.

23

Version-Space



24

Version-Space Learning Algorithm

- Initially, G-Set = TRUE (everything) and S-Set = FALSE (∅)
- For each example e, Update Version-Space

1. *False positive for S_i*

   S_i is too general, so we throw it out of the S-Set.

2. *False negative for S_i*

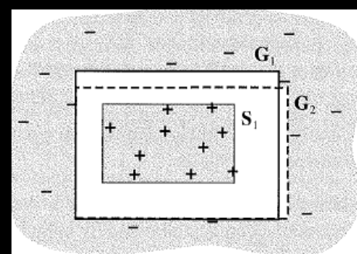
   S_i is too specific, so we replace it by its immediate generalization.

3. *False positive for G_i*

   G_i is too general, so we replace it by its immediate specialization.

4. *False negative for G_i*

   G_i is too specific, so we throw it out of the G-Set.



25

Terminal conditions for Version-Space Learning Algorithm

Continue update operations until one of three things happens:

1. We have exactly one concept left in version space: the unique hypothesis
2. The version space collapses, i.e, either S-Set or G-Set = ∅:
   no consistent hypotheses for training set. - the failure of DT alg.
3. We run out of examples but still remaining hypotheses: VS= $\vee_i\, h_i$s.
   For any new example,
   return their classification of the example if all the h_i agree.
   take the majority vote, if they disagree.

26

Drawbacks of Version-Space Learning Algorithm

1. If the domain contains noise or insufficient attributes, the VS always collapse.
2. It we allow unlimited disjunction in the hypothesis space,
   the S-set will always contain a single most-specific hypothesis
   (i.e. disjunction of the description of the positive example)
   similarly, the G-set will contain a single most-general hypothesis
   (i.e. the negation of disjunction the description of the negative example).
3. For some HS, |G/S-set| may grow exponentially in the # of attributes.

A possible solution for the problem of disjunction:

allow only limited forms of disjunction

Include a generalization hierarchy of more general predicates:

e.g.) $WaitEstimate(x, 30\text{-}60) \vee WaitEstimate(x, >60) \rightarrow LongWait(x)$

- The 1st Application of pure version space algorithm: meta-Dendral system
  - Learning of rules for predicting the breakage of molecules.