LOGICAL LEARNING

Outline

- Logical Formulation of Learning
- Current-Best-Hypothesis Search
- Least-Commitment Search

Based on the textbook by Stuart Russell & Peter Norvig:

Artificial Intelligence, A Modern Approach (2nd Edition)

Section 19.1

1. LOGICAL FORMULATION OF LEARNING

- Inductive learning was previously defined as a process of searching for a hypothesis that agrees with the observed examples.
- For now we concentrate on the case where hypotheses, examples, classifications are **represented** in terms of *logical sentences*.
- This form of learning is more general and complex compared to learning decision trees or decision lists.
- This approach allows for *incremental construction* of hypotheses, one sentence at a time—allowing for **prior knowledge**, too.
- The full power of logical inference can be utilized in learning.

Examples and Hypotheses

- In the logical representation, attributes become unary predicates.
- The i^{th} example is denoted by X_i and its description by $D_i(X_i)$.
- The generic notations $Q(X_i)$ and $\neg Q(X_i)$ are used for *positive* and *negative* examples, respectively.
- The complete training set corresponds to the conjunction of the respective description and classification sentences.

Example. The first example in the restaurant domain is described by the following logical sentence:

$$Alternate(X_1) \land \neg Bar(X_1) \land \neg Fri/Sat(X_1) \land Hungry(X_1) \land \ldots$$

The classification of X_1 is given by $WillWait(X_1)$.

Restaurant Domain Revisited

- Recall the 12 examples generated from Mr. Russell's decision tree:

Example	Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0–10	Yes
X_2	Yes	No	No	Yes	Full	$	No	No	Thai	30–60	No
X_3	No	Yes	No	No	Some	$	No	No	Burger	0–10	Yes
X_4	Yes	No	Yes	Yes	Full	$	No	No	Thai	10–30	Yes
X_5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	No
X_6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0–10	Yes
X_7	No	Yes	No	No	None	$	Yes	No	Burger	0–10	No
X_8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0–10	Yes
X_9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	No
X_{10}	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10–30	No
X_{11}	No	No	No	No	None	$	No	No	Thai	0–10	No
X_{12}	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30–60	Yes

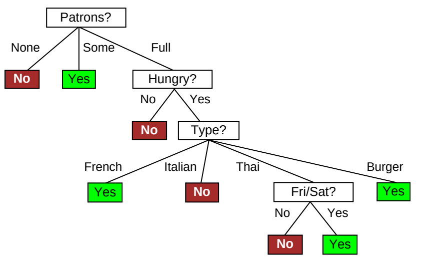
Candidate Definitions

- The aim is to find an equivalent logical expression for the goal predicate Q that can be used to classify examples correctly.
- Each hypothesis H_i proposes a **candidate definition** $C_i(x)$ for the goal predicate Q, i.e. H_i takes the form $\forall x(Q(x) \leftrightarrow C_i(x))$.
- The **extension** of a hypothesis $H_i = \forall x(Q(x) \leftrightarrow C_i(x))$ is the set of examples X for which $Q(X)$ evaluates to true.

Example. In the restaurant domain, the extension of the hypothesis $\forall r(WillWait(r) \leftrightarrow Patrons(r, Some))$ includes, e.g., X_1, X_3, X_6, and X_8.

But this does not match with the intended meaning of $WillWait(X)$!

Example



The decision tree above corresponds to the following description:

$$
\begin{aligned}
H_1 = \forall r(WillWait(r) \leftrightarrow & Patrons(r, Some) \lor \\
& (Patrons(r, Full) \land Hungry(r) \land Type(r, French)) \lor \\
& (Patrons(r, Full) \land Hungry(r) \land Type(r, Thai) \land Fri/Sat(r)) \lor \\
& (Patrons(r, Full) \land Hungry(r) \land Type(r, Burger))\ )
\end{aligned}
$$

Hypothesis Space

- Logically equivalent hypotheses have equal extensions.
- Two hypotheses with different extensions are logically inconsistent with each other, as they differ on at least one example X_i.
- The hypothesis space $\{H_1, H_2, \ldots, H_n\}$ is denoted by $\mathbf{H}$.
- It is usually believed that one of the hypotheses in $\mathbf{H}$ is correct, i.e. the disjunction $H_1 \lor H_2 \lor \ldots \lor H_n$ evaluates to true.
- In decision tree learning, the hypothesis space consists of all decision trees definable in terms of the attributes provided.

Example. The conjunction of $H_2 = \forall r(WillWait(r) \leftrightarrow Hungry(r))$ and $H_3 = \forall r(WillWait(r) \leftrightarrow \neg Hungry(r))$ implies a contradiction.

Classifying Examples with Hypotheses

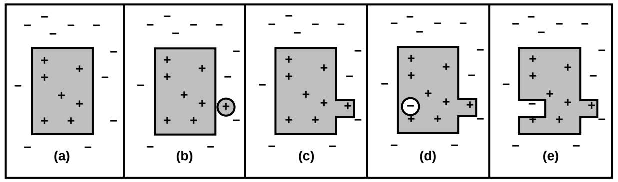
- Given a hypothesis $H_i = \forall x(Q(x) \leftrightarrow C_i(x))$, an example X is **positive/negative** if $Q(X)/\neg Q(X)$ evaluates to true.
- A **false** positive/negative example X for a hypothesis $H_i = \forall x(Q(x) \leftrightarrow C_i(x))$ gets an incorrect classification by H_i.
- Inductive learning can be seen as a process of gradually eliminating hypotheses that are inconsistent with examples.

Example. For H_1 in the restaurant domain, the first example X_1 is a positive one, as $WillWait(X_1)$ evaluates to true.

On the other hand, X_1 is a false negative example for $H_3 = \forall r(WillWait(r) \leftrightarrow \neg Hungry(r))$, as $Hungry(X_1)$ holds.

2. CURRENT-BEST-HYPOTHESIS SEARCH

- The idea is to maintain a single hypothesis H, and to adjust it as new examples arrive in order to maintain consistency.
- The current hypothesis H is illustrated in the figure (a) below.
- A false negative example (b) can be removed by a **generalization** (c) that extends the extension of the current hypothesis H_i.
- A false positive example (d) can be removed by a **specialization** (e) that narrows the extension of the current hypothesis H_i.



Skeletal Algorithm

Current-best-hypothesis search is captured by the following algorithm:

```
function CURRENT-BEST-LEARNING(examples) returns a hypothesis

  H ← any hypothesis consistent with the first example in examples
  for each remaining example in examples do
     if e is false positive for H then
        H ← choose a specialization of H consistent with examples
     else if e is false negative for H then
        H ← choose a generalization of H consistent with examples
     if no consistent specialization/generalization can be found then fail
  end
  return H
```

- Generalizations and specializations imply *logical relationships*:

  E.g., if $H_1 = \forall x(Q(x) \leftrightarrow C_1(x))$ is a generalization of $H_2 = \forall x(Q(x) \leftrightarrow C_2(x))$, then $\forall x(C_2(x) \rightarrow C_1(x))$ holds.
- Note that H_2 is a specialization of H_1 in the setting above.

Example I

- A way to generalist is to **drop conditions** from hypotheses.
- E.g., the hypothesis $\forall x(\mathit{WillWait}(x) \leftrightarrow \mathit{Patrons}(x,\mathit{Some}))$ generalizes $\forall x(\mathit{WillWait}(x) \leftrightarrow \mathit{Alternate}(x) \wedge \mathit{Patrons}(x,\mathit{Some}))$.

Example	Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0–10	Yes
X_2	Yes	No	No	Yes	Full	$	No	No	Thai	30–60	No
X_3	No	Yes	No	No	Some	$	No	No	Burger	0–10	Yes
X_4	Yes	No	Yes	Yes	Full	$	No	No	Thai	10–30	Yes
X_5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	No
X_6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0–10	Yes
X_7	No	Yes	No	No	None	$	Yes	No	Burger	0–10	No
X_8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0–10	Yes
X_9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	No
X_{10}	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10–30	No
X_{11}	No	No	No	No	None	$	No	No	Thai	0–10	No
X_{12}	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30–60	Yes

Example II

Example. Hypotheses are formed in the restaurant example as follows:

H_1: $\forall x(\mathit{WillWait}(x) \leftrightarrow \mathit{Alternate}(x))$

H_2: $\forall x(\mathit{WillWait}(x) \leftrightarrow \mathit{Alternate}(x) \wedge \mathit{Patrons}(x,\mathit{Some}))$

H_3: $\forall x(\mathit{WillWait}(x) \leftrightarrow \mathit{Patrons}(x,\mathit{Some}))$

H_4: $\forall x(\mathit{WillWait}(x) \leftrightarrow \mathit{Patrons}(x,\mathit{Some}) \vee (\mathit{Patrons}(x,\mathit{Full}) \wedge \mathit{Fri/Sat}(x)))$

There are also other hypotheses conforming to the first four examples:

H_4': $\forall x(\mathit{WillWait}(x) \leftrightarrow \neg \mathit{WaitEstimate}(x,\mathit{30\text{-}60}))$

H_4'': $\forall x(\mathit{WillWait}(x) \leftrightarrow \mathit{Patrons}(x,\mathit{Some}) \vee$
$(\mathit{Patrons}(x,\mathit{Full}) \wedge \mathit{WaitEstimate}(x,\mathit{10\text{-}30}))\ )$

Discussion

- The CURRENT-BEST-LEARNING algorithm is *non-deterministic*: there may be several possible specializations or generalizations that can be applied at any point.
- The choices made might not lead to the simplest hypothesis.
- If a dead-end (unrecoverable inconsistency) is encountered, the algorithm must backtrack to a previous choice point.
- Checking the consistency of all the previous examples over again for each choice is very expensive.

3. LEAST-COMMITMENT SEARCH

- The original hypothesis space can be seen as a huge disjunction

$$H_1 \vee H_2 \vee \ldots \vee H_n.$$

- Hypotheses which are consistent with all examples encountered so far form a set of hypotheses called the **version space** V.
- Version space is shrunk by the **candidate elimination** algorithm:

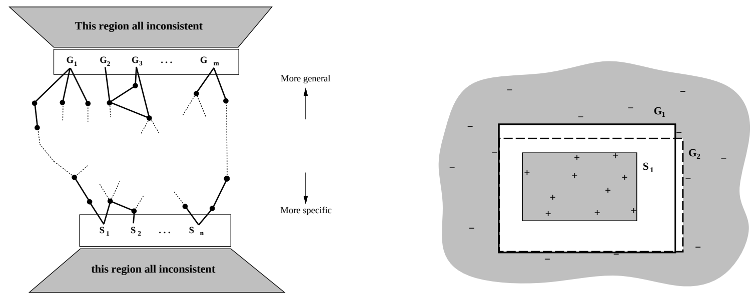
```
function VERSION-SPACE-LEARNING(examples) returns a version space
  local variables: V, the version space: the set of all hypotheses

  V ← the set of all hypotheses
  for each example e in examples do
     if V is not empty then V ← VERSION-SPACE-UPDATE(V, e)
  end
  return V
---------------------------------------------------------------
function VERSION-SPACE-UPDATE(V, e) returns an updated version space
  V ← {h ∈ V : h is consistent with e}
```


Boundary Sets

- The algorithm finds a subset of the version space V that is consistent with all examples in an *incremental* way.
- Candidate elimination is an example of a **least-commitment** algorithm, as no arbitrary choices are made among hypotheses.
- Since the hypothesis space V is possibly enormous, it cannot be represented directly as a set of hypotheses or a disjunction.
- The problem can be alleviated by **boundary sets** $\{S_1, \ldots, S_n\}$ (**S-set**) and $\{G_1, \ldots, G_m\}$ (**G-set**) and a partial ordering among hypotheses induced by specialization/generalization.
- Any hypothesis H between a most specific boundary S_i and a most general boundary G_j is consistent with the examples seen.

Illustration of Boundary Sets



- Initially, the S-set contains a single hypothesis $\forall x(Q(x) \leftrightarrow False)$ while the G-set contains $\forall x(Q(x) \leftrightarrow True)$ only.
- The remaining problem is how to update S-sets and G-sets for a new example (the job of the VERSION-SPACE-UPDATE function).

Updating Version Space

- Upon a false negative/positive example, a most specific boundary S is replaced by all its immediate generalizations / deleted.
- Upon a false positive/negative example, a most general boundary G is replaced by all its immediate specializations / deleted.

These operations on S-sets and G-sets are continued until:

1. There is exactly one hypothesis left in the version space.
2. The version space *collapses* (i.e., the S-set or G-set becomes empty): there are no consistent hypotheses for the training set.
3. We run out of examples with several hypotheses remaining in the version space: a solution is to take the majority vote.

Discussion

- If the domain contains noise or insufficient attributes for exact classification, the version space will always collapse.
- If unlimited disjunction is allowed in the hypothesis space, the S-set will always contain a single most-specific hypothesis (disjunction of positive examples seen to date).
- Analogously for the G-set and negative examples.
- A solution is to allow only limited forms of disjunction.
- For certain hypothesis spaces, the number of elements in the S-set and G-set may grow exponentially in the number of attributes.

SUMMARY

- Learning is essential for dealing with unknown environments.
- In **cumulative learning**, a learning agent improves its ability to learn as it acquires more knowledge.
- Prior knowledge helps learning by eliminating otherwise consistent hypotheses and by “filling in” the explanation of examples, thereby allowing for shorter hypotheses.