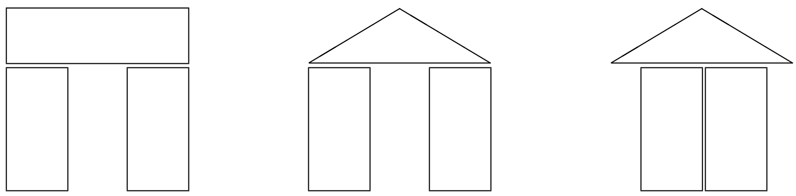
Concept Learning

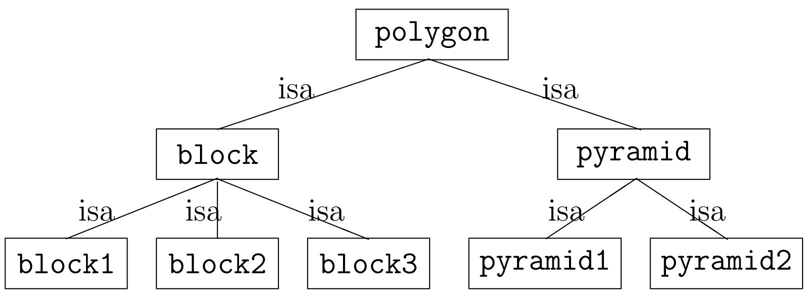
1 Learning the concept of arch

Examples



*Example*1 *Example*2 *Near_arch*

Background knowledge



2 Using semantic nets

$Example1 =$
$\{partof(block1, arch), partof(block2, arch), partof(block3, arch),$
$supports(block1, block3), supports(block2, block3)\}$

$Example2 =$
$\{partof(block1, arch), partof(block2, arch), partof(pyramid1, arch),$
$supports(block1, pyramid1), supports(block2, pyramid1)\}$

$Apriori\_knowledge =$
$\{isa(block1, block), isa(block2, block), isa(block3, block),$
$isa(block, polygon), isa(pyramid1, pyramid), isa(pyramid, polygon)\}$

3 Generalization

$$Example1 + Example2 \Rightarrow Hypothesis1$$

$Hypothesis1 =$
$\{partof(block1, arch), partof(block2, arch), partof(polygon, arch),$
$supports(block1, polygon), supports(block2, polygon)\}$

4 Specialization

$$Hypothesis1 + Near\_miss \Rightarrow Hypothesis2$$

$Near\_miss =$
$\{partof(block1, arch), partof(block2, arch), partof(polygon, arch),$
$supports(block1, polygon), supports(block2, polygon),$
$touches(block1, block2)\}$

$Hypothesis2 =$
$\{partof(block1, arch), partof(block2, arch), partof(polygon, arch),$
$supports(block1, polygon), supports(block2, polygon),$
$doesnottouches(block1, block2)\}$

5 Issues

- First concept learning system (Winston, 1975)
- Incremental learning
- Order of examples is important

6 Induction task

Formal system: language L (with three subsets – L_B (*language of background knowledge*), L_E (*language of examples*) and L_H (*language of hypotheses*), and *a derivibilty relation* "$\rightarrow$" – a mapping between elements from L. Example: First-Order Logic (Predicate calculus).

Induction task: Given *background knowledge* $B \in L_B$, *positive examples* $E^+ \in L_E$ and *negative examples* $E^- \in L_E$, find a hypothesis $H \in L_H$, such that:

1. $B \not\rightarrow E^+$ (*nessecity*);
2. $B \not\rightarrow E^-$ (*consistency of B*);
3. $B \cup H \rightarrow E^+$ (*sufficiency*);
4. $B \cup H \not\rightarrow E^-$ (*consistency of H*).

Straightforward solution: $H = E^+$, however:

- No new examples accepted (no induction step).
- No explanation of E^+ in terms of B.

Anyway, $H = E^+$ is useful for searching the *hypothesis space.*
$H = E^+$ is called *most specific* hypothesis, denoted $\perp$.

7 Generality/specificity

Generality (subsumption, coverage) of hypotheses. Let H and H' be hypotheses, where $H \rightarrow E$ and $H' \rightarrow E'$. *H is more general than (subsumes, covers) H'*, denoted $H \geq H'$, if $E \supseteq E'$.

Semantic ordering. Ordering of hypotheses is based on coverage of examples.

Most general hypothesis $\top$. A hypothesis that covers all examples from L_E.

- Easy to find for any particular language.
- However, $\perp$ does not satisfy the conditions of the induction task (covers E^-).
- Even if $E^- = \emptyset$, $\top$ is not suitable either, because it is not constructive.

Hypothesis space. All hypotheses H, such that $\top \geq H \geq \perp$.

Generalization/specialization operators. Procedures (algorithms) that given a hypothesis H genarate a new hypothesis H' that is more general/specific than H.

Example of a hypotesis space. Power set of E (2^E).

- Lattice structure induced by the *subset* ($\subseteq$) relation (for every two elements a least upper bound exists).
- Assume that a hypothesis can be identified for every subset of E.
- Then the hypothesis space can be easily searched (lattices are well studied algebraic structures with a lot of nice properties).

However,

- Every hypothesis can be associated with a set of examples, but the reverse is not generally true.
- In more complex languages (e.g. First-Order Logic) constructive operators for generalization/specialization cannot be found or, if found, are non-computable.

Therefore we mostly use

Syntactic orderings. Orderings that are determined directly by the representation language.

- Syntactic orderings are usually stronger (i.e. they hold for fewer objects) than the semantic ones.
- Consequently syntactic orderings are *incomplete* – they do not guarantee exhaustive search in the hypothesis space.
- This, in turn, may cause to skip over the desired hypothesis and generate a hypothesis that is either too specific or too general. These problems are known as *overspecialization* and *overgerenalization*.

8 Criteria for choosing generalization/specialization operators

- The languages of examples and hypotheses (the so called *syntactic* or *language bias*);
- The strategy for searching the hypothesis space (*search bias*);
- The criteria for hypothesis evaluation and selection.

9 Annotated example

Language (used both for examples and hypotheses): all subsets of $\{a, b, c, d\}$

Generality ordering (covering): *subset* ($\subseteq$) relation. That is, $H \geq H'$, if $H \subseteq H'$.

Generalization operation: dropping element.

- For example, let $H = \{a, b\}$. If we drop b from this set, we get $H' = \{a\}$, which is more general, because $\{a\} \subseteq \{a, b\}$.
- The most general element in this language ($\top$) is the empty set $\{\}$ (it's subset of all other sets).
- The most specific element $\bot = \{a, b, c, d\}$, because all other sets are subsets of $\{a, b, c, d\}$.

Assume we have:

- Two positive examples, $E^+ = \{\{a, b, c\}, \{a, b, d\}\}$.
- One negative example, $E^- = \{c\}$.

Let's consier the following two hypoteses: $H_1 = \{a, b\}$, $H_2 = \{d\}$

- H_1 is a good hypothesis, because it covers all positives and none negatives.
- H_2 is not as good as H_1, because it covers just one positive ($\{a, b, d\}$), i.e. it is *incomplete*. Still, H_2 is *correct* as it does not cover the negative example $\{c\}$.
- Semantically H_1 is more general than H_2, because H_1 covers two examples (it's subset of both $\{a, b, c\}$ and $\{a, b, d\}$) and H_2 covers one.

- However, there is no syntactic relation between H_1 and H_2. That is, there is no subset relation between them. Consequently, if we are searching the hypothesis space by applying generalization operations (dropping elements) and start from H_2 (which is more specific) we cannot reach H_1 (the more general one).
- If we generalize H_2, we get $\top = \{\}$, which covers all examples, i.e. it's both syntactically and semantically more general and H_2. However, this is not a good hypothesis, because along with the positives, it covers negatives too ($\{c\}$). This is an example of *over-generalization*.

Despite the problems with the syntactic ordering, we still can use it to find the best hypothesis. For example:

- Starting the search from $\top = \{\}$ we have to follow the paths:
  $\{\} \rightarrow \{a\} \rightarrow \{a, b\}$ or $\{\} \rightarrow \{b\} \rightarrow \{a, b\}$.
- Starting the search from $\bot = \{a, b, c, d\}$ we have to follow the path:
  $\{a, b, c, d\} \rightarrow \{a, b, c\} \rightarrow \{a, b\}$.