Candidate Elimination Algorithm [Mitchell 78]

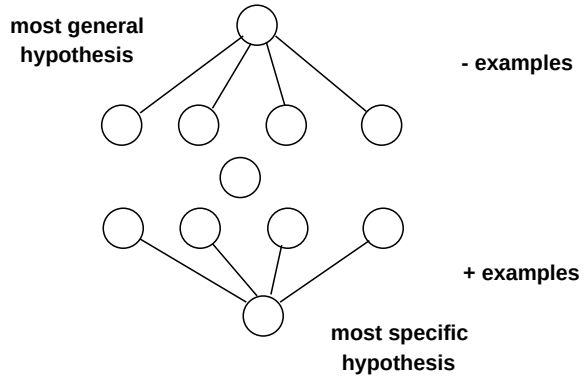
Version Space Method

- Assumes f is a Boolean function.
- Requires noise-free positive and negative examples.
- Assumes that the concept can be described in terms of a conjunction of the available attributes. (No negation.)

Algorithm maintains a **version space** that keeps track of all concept descriptions, H, consistent with the training instances without remembering any of the instances. Processes the instances incrementally.

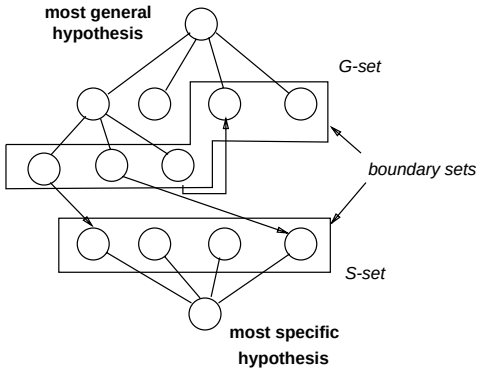
Learning in a Version Space

Key idea: Generalization of the specific concept descriptions and specialization of the general concept descriptions ultimately leads to just one concept description.



Details

Each specialization must be a generalization of some specific concept description. No specialization can be a specialization of another general concept description.



Example

Num	Restaurant	Meal	Day	Cost	Reaction
1	The Nines	bkfst	Fri	$	sick (+)
2	Banfis	lunch	Fri	$$	ok (-)
3	The Nines	lunch	Sat	$	sick (+)
4	Moosewood	bkfst	Sun	$	ok (-)
5	The Nines	bkfst	Sun	$$	ok (-)

Generalization and Specialization Operators

Specialization: replace one "?" with a value

```
[? ? ? ?]
```

```
[9s ? ? ?][Banfis ? ? ?]...[? bkfst ? ?][? lunch ? ?]...
```

Generalization: replace one value with "?"

```
[? bkfst Fri $][9s ? Fri $][9s bkfst ? $][9s bkfst Fri ?]
```

```
[9s bkfst Fri $]
```


Algorithm

Initialize S = , G =
Get next training instance, I. If I is +, then:

1. Retain in G only those descriptions that match I.
2. Generalize members of S that do not match I, only to the extent required to allow them to match I, producing a new set S'. (S' = matching S members plus new generalizations.)
3. Remove from S' all members that are more general than some other member of S'.
4. Remove from S' all members that aren't at least as specific as some member of G.
5. Set S to S'.

If I is -, then:

1. Retain in S only those descriptions that do *not* match I.
2. Specialize members of G that match I, only to the extent required to keep them from matching I, producing a new set G'. (G' = non-matching G members plus new specializations.)
3. Remove from G' all members that are more specific than some other member of G'.
4. Remove from G' all members that aren't at least as general as some member of S.
5. Set G to G'.

Example

```
[? ? ? ?]
```

```
[9s bkfst Fri $]
```