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.

Example

<table>
<thead>
<tr>
<th>Num</th>
<th>Restaurant</th>
<th>Meal</th>
<th>Day</th>
<th>Cost</th>
<th>Reaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The Nines</td>
<td>bkfst</td>
<td>Fri</td>
<td>$</td>
<td>sick (+)</td>
</tr>
<tr>
<td>2</td>
<td>Banfis</td>
<td>lunch</td>
<td>Fri</td>
<td>$$</td>
<td>ok (-)</td>
</tr>
<tr>
<td>3</td>
<td>The Nines</td>
<td>lunch</td>
<td>Sat</td>
<td>$</td>
<td>sick (+)</td>
</tr>
<tr>
<td>4</td>
<td>Moosewood</td>
<td>bkfst</td>
<td>Sun</td>
<td>$</td>
<td>ok (-)</td>
</tr>
<tr>
<td>5</td>
<td>The Nines</td>
<td>bkfst</td>
<td>Sun</td>
<td>$$</td>
<td>ok (-)</td>
</tr>
</tbody>
</table>
Generalization and Specialization Operators

**Specialization:** replace one “?” with a value

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

```
```

**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 $]
```