Introduction to Machine Learning

CSE474/574: Lecture 4

Varun Chandola <chandola@buffalo.edu>

2 Feb 2015

Outline

Outline

Compressing Version Space

More_General_Than Relationship

$$h_j \geq_g h_k \quad if \quad h_k(x) = 1 \Rightarrow h_j(x) = 1$$
$$h_j >_g h_k \quad if \quad (h_j \geq_g h_k) \wedge (h_k \not\geq_g h_j)$$

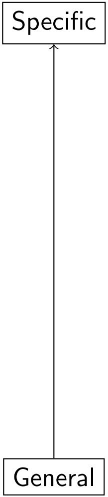
- In a version space, there are:
  1. Maximally general hypotheses
  2. Maximally specific hypotheses
- Boundaries of the version space

Example

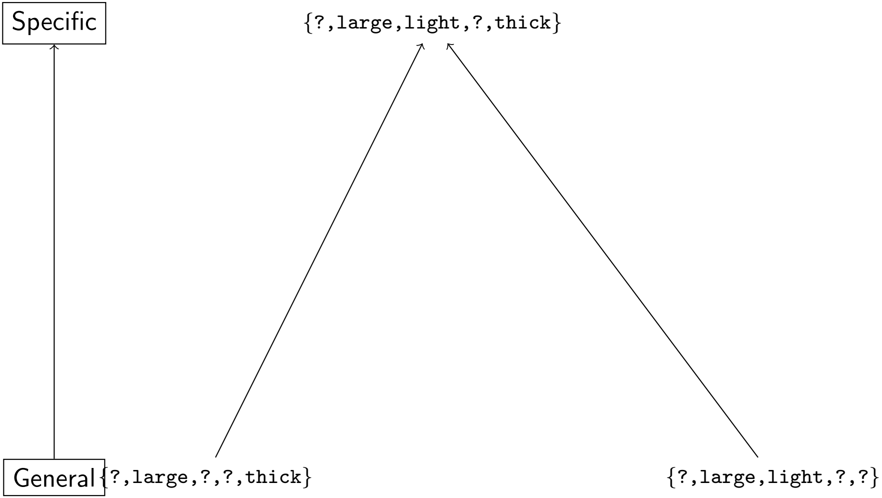
1. {circular,large,light,smooth,thick}, malignant
2. {circular,large,light,irregular,thick}, malignant
3. {oval,large,dark,smooth,thin}, benign
4. {oval,large,light,irregular,thick}, malignant
5. {circular,small,light,smooth,thick}, benign

- {?,large,light,?,thick}
- {?,large,?,?,thick}
- {?,large,light,?,?}

Example (2)



Example (2)


Specific
{?,large,light,?,thick}
General
{?,large,?,?,thick}
{?,large,light,?,?}

Boundaries are Enough to Capture Version Space

Version Space Representation Theorem

Every hypothesis h in the version space is *contained within* at least one pair of hypothesis, g and s, such that $g \in G$ and $s \in S$, i.e.,:

$$g \geq_g h \geq_g s$$

Candidate Elimination Algorithm

1. Initialize $S_0 = \{\emptyset\}$, $G_0 = \{?, ?, \ldots, ?\}$
2. For every training example, $d = \langle x, c(x) \rangle$

$c(x) = +ve$

1. Remove from G any g for which $g(x) \neq +ve$
2. For every $s \in S$ such that $s(x) \neq +ve$:
    1. Remove s from S
    2. For every *minimal generalization*, s' of s
        - If $s'(x) = +ve$ and there exists $g' \in G$ such that $g' >_g s'$
        - Add s' to S
3. Remove from S all hypotheses that are more general than another hypothesis in S

$c(x) = -ve$

1. Remove from S any s for which $s(x) \neq -ve$
2. For every $g \in G$ such that $g(x) \neq -ve$:
    1. Remove g from G
    2. For every *minimal specialization*, g' of g
        - If $g'(x) = -ve$ and there exists $s' \in S$ such that $g' >_g s'$
        - Add g' to G
3. Remove from G all hypotheses that are more specific than another hypothesis in G

Candidate Elimination Algorithm

1. Initialize $S_0 = \{\emptyset\}$, $G_0 = \{?, ?, \ldots, ?\}$
2. For every training example, $d = \langle x, c(x) \rangle$

$c(x) = +ve$

1. Remove from G any g for which $g(x) \neq +ve$
2. For every $s \in S$ such that $s(x) \neq +ve$:
    1. Remove s from S
    2. For every *minimal generalization*, s' of s
        - If $s'(x) = +ve$ and there exists $g' \in G$ such that $g' >_g s'$
        - Add s' to S
3. Remove from S all hypotheses that are more general than another hypothesis in S

$c(x) = -ve$

1. Remove from S any s for which $s(x) \neq -ve$
2. For every $g \in G$ such that $g(x) \neq -ve$:
    1. Remove g from G
    2. For every *minimal specialization*, g' of g
        - If $g'(x) = -ve$ and there exists $s' \in S$ such that $g' >_g s'$
        - Add g' to G
3. Remove from G all hypotheses that are more specific than another hypothesis in G

Candidate Elimination Algorithm

1. Initialize $S_0 = \{\emptyset\}$, $G_0 = \{?, ?, \ldots, ?\}$
2. For every training example, $d = \langle x, c(x) \rangle$

$c(x) = +ve$

1. Remove from G any g for which $g(x) \neq +ve$
2. For every $s \in S$ such that $s(x) \neq +ve$:
    1. Remove s from S
    2. For every *minimal generalization*, s' of s
        - If $s'(x) = +ve$ and there exists $g' \in G$ such that $g' >_g s'$
        - Add s' to S
3. Remove from S all hypotheses that are more general than another hypothesis in S

$c(x) = -ve$

1. Remove from S any s for which $s(x) \neq -ve$
2. For every $g \in G$ such that $g(x) \neq -ve$:
    1. Remove g from G
    2. For every *minimal specialization*, g' of g
        - If $g'(x) = -ve$ and there exists $s' \in S$ such that $g' >_g s'$
        - Add g' to G
3. Remove from G all hypotheses that are more specific than another hypothesis in G

Example

Specific S_0 $\{\emptyset\}$

General G_0 $\{?,?,?,?,?\}$

Example

Specific S_0 $\{\emptyset\}$

S_1 $\{\texttt{ci,la,li,sh,th}\}$

$\langle\{\texttt{ci,la,li,sh,th}\}, +\text{ve}\rangle$

General G_0 $\{\texttt{?,?,?,?,?}\}$

Example

Specific S_0 $\{\emptyset\}$

S_1 $\{\texttt{ci,la,li,sh,th}\}$

$\langle\{\texttt{ci,la,li,sh,th}\}, +\text{ve}\rangle$

$\langle\{\texttt{ci,la,li,ir,th}\}, +\text{ve}\rangle$

S_2 $\{\texttt{ci,la,li,?,th}\}$

General G_0 $\{\texttt{?,?,?,?,?}\}$

Example

Specific S_0 $\{\emptyset\}$

S_1 $\{\mathtt{ci,la,li,sh,th}\}$

S_2 $\{\mathtt{ci,la,li,?,th}\}$

$\langle\{\mathtt{ci,la,li,sh,th}\}, +\mathrm{ve}\rangle$

$\langle\{\mathtt{ci,la,li,ir,th}\}, +\mathrm{ve}\rangle$

$\langle\{\mathtt{ov,sm,li,sh,tn}\}, \text{-ve}\rangle$

G_3 $\{\mathtt{ci,?,?,?,?}\},\{\mathtt{?,la,?,?,?}\},\{\cancel{\mathtt{?,?,dk,?,?}}\},\{\cancel{\mathtt{?,?,?,ir,?}}\},\{\mathtt{?,?,?,?,th}\}$

General G_0 $\{\mathtt{?,?,?,?,?}\}$

Example

Specific S_0 $\{\emptyset\}$

S_1 $\{\texttt{ci,la,li,sh,th}\}$

S_2 $\{\texttt{ci,la,li,?,th}\}$

S_4 $\{\texttt{?,la,li,?,th}\}$

$\langle\{\texttt{ci,la,li,sh,th}\}, +\text{ve}\rangle$

$\langle\{\texttt{ci,la,li,ir,th}\}, +\text{ve}\rangle$

$\langle\{\texttt{ov,sm,li,sh,tn}\}, \text{-ve}\rangle$

$\langle\{\texttt{ov,la,li,ir,th}\}, +\text{ve}\rangle$

G_3 $\{\cancel{\texttt{ci,?,?,?,?}}\},\{\texttt{?,la,?,?,?}\},\{\texttt{?,?,?,?,th}\}$

G_3 $\{\texttt{ci,?,?,?,?}\},\{\texttt{?,la,?,?,?}\},\{\cancel{\texttt{?,?,dk,?,?}}\},\{\cancel{\texttt{?,?,?,ir,?}}\},\{\texttt{?,?,?,?,th}\}$

General G_0 $\{\texttt{?,?,?,?,?}\}$

Understanding Candidate Elimination

- S and G boundaries move towards each other
- Will it converge?
    1. No errors in training examples
    2. Sufficient training data
    3. The target concept is in $\mathcal{H}$
- Why is it better than *Find-S*?

Not Sufficient Training Examples

- Use boundary sets S and G to make predictions on a new instance x^*
- **Case 1**: x^* is **consistent** with every hypothesis in S
- **Case 2**: x^* is **inconsistent** with every hypothesis in G

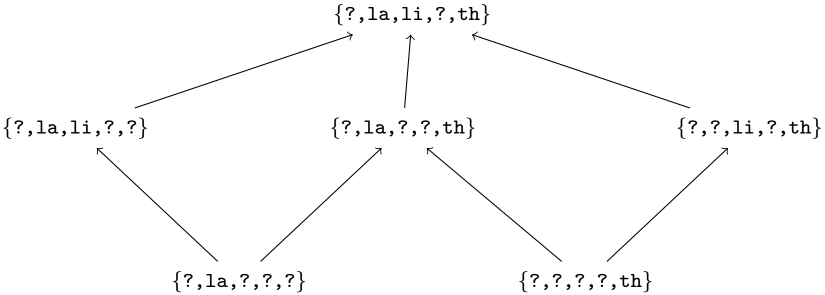
Partially Learnt Concepts - Example

{?,la,li,?,th}

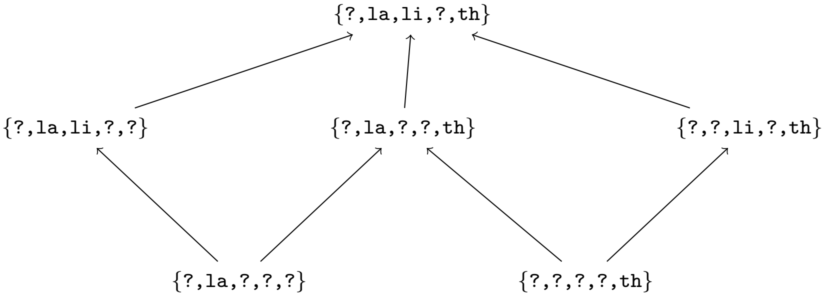
{?,la,?,?,?}

{?,?,?,?,th}

Partially Learnt Concepts - Example

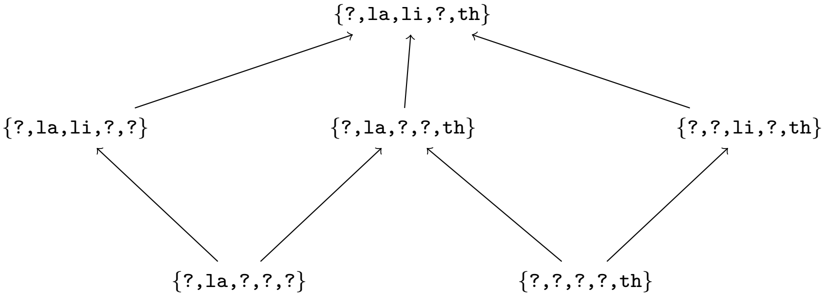


Partially Learnt Concepts - Example



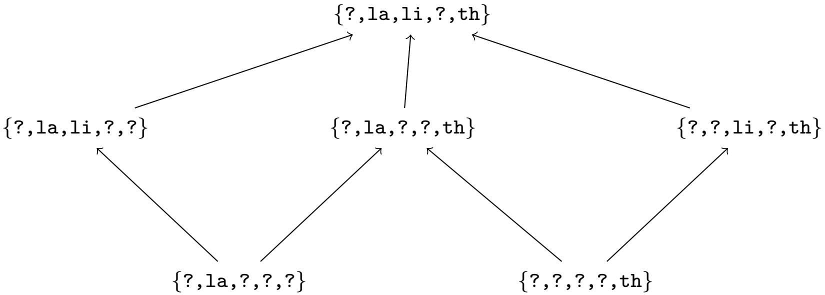
{ci,la,li,sh,th}, **?**

Partially Learnt Concepts - Example



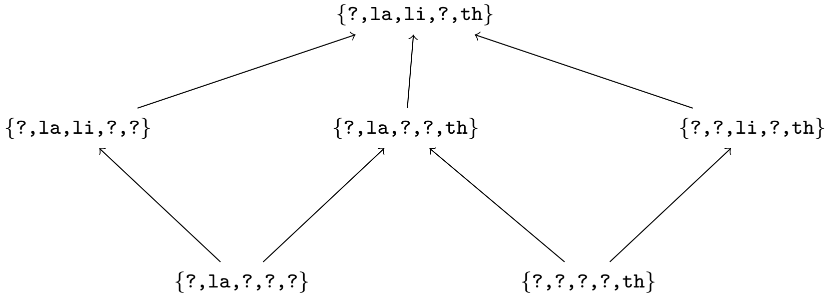
{ov,sm,li,ir,tn}, **?**

Partially Learnt Concepts - Example



{ov,la,dk,ir,th}, **?**

Partially Learnt Concepts - Example



{ci,la,li,ir,tn}, **?**

Using Partial Version Spaces

- **Halving Algorithm**
  - Predict using the majority of concepts in the version space
- **Randomized Halving Algorithm** [1]
  - Predict using a randomly selected member of the version space

References

W. Maass.
On-line learning with an oblivious environment and the power of randomization.
In *Proceedings of the Fourth Annual Workshop on Computational Learning Theory*, COLT '91, pages 167–178, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc.