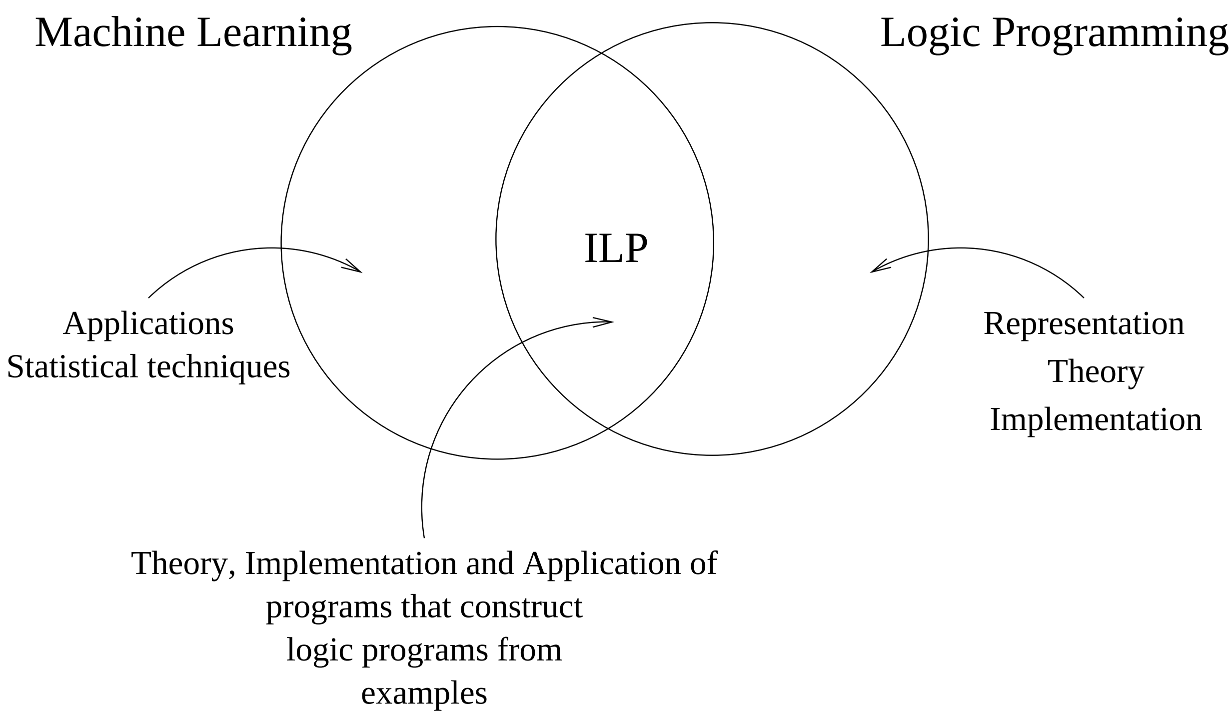
What is ILP?

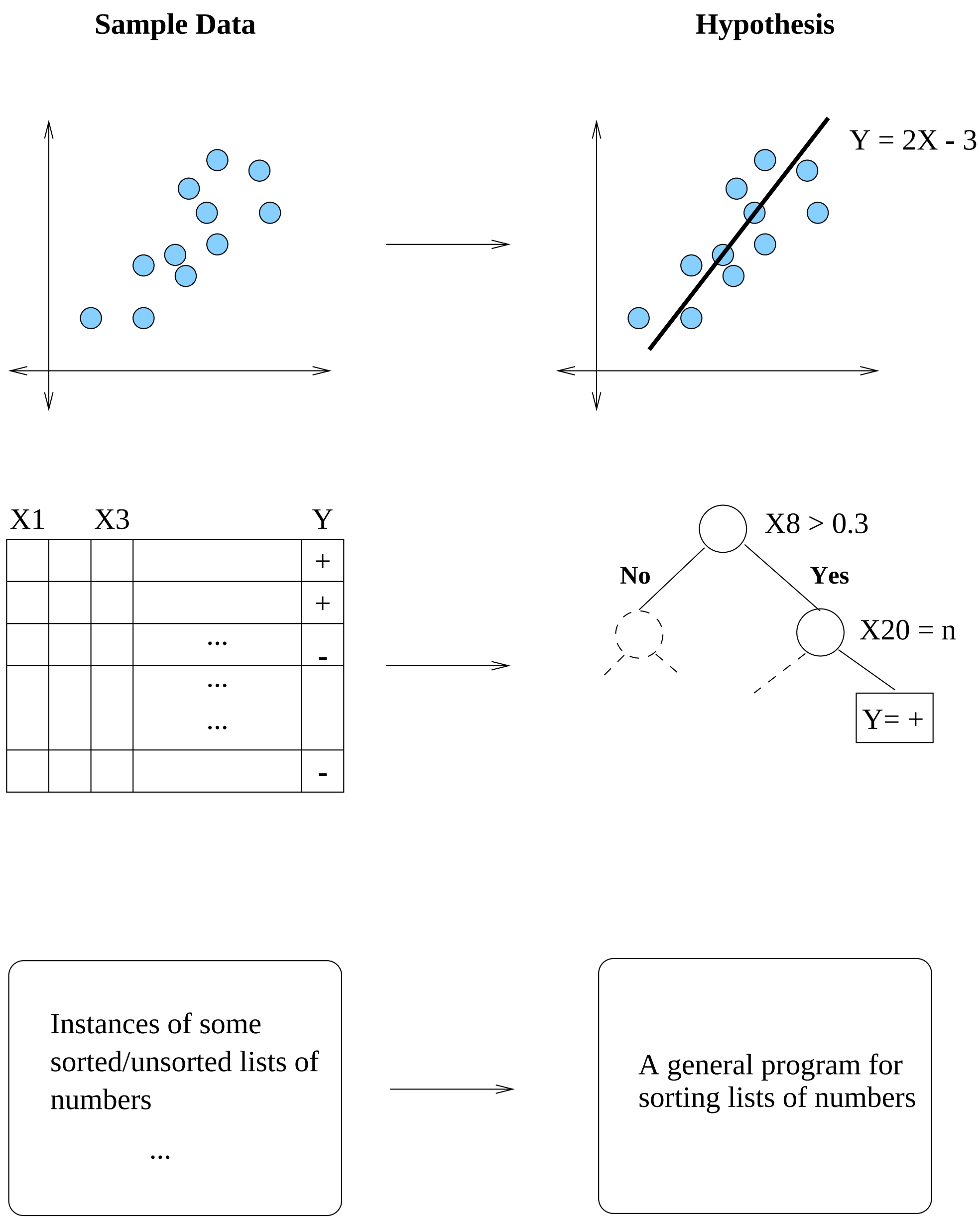
Inductive Logic Programming $\times$

Inductive Logic Programming $\surd$



Machine Learning

Programs that hypothesize general descriptions from sample data



Logic Programming

Study of using symbolic logic as a programming language

Specification = Programming

Logic program:
$\forall X, Y$ grandfather$(X, Y) \leftarrow \exists Z$ (father(X, Z), parent(Z, Y))
father(henry,jane) $\leftarrow$
father(henry,joe) $\leftarrow$
parent(jane,john) $\leftarrow$
parent(joe,robert) $\leftarrow$

Derived facts:
grandfather(henry,john) $\leftarrow$
grandfather(henry,robert) $\leftarrow$

"Inductive" Logic Programming

(Sample data)

Examples:
grandfather(henry,john) $\leftarrow$
grandfather(henry,robert) $\leftarrow$

$$+$$

Background:
father(henry,jane) $\leftarrow$
father(henry,joe) $\leftarrow$
parent(jane,john) $\leftarrow$
parent(joe,robert) $\leftarrow$

$$\downarrow$$

Hypothesis:
$\forall X, Y$ grandfather$(X, Y) \leftarrow \exists Z$ (father(X, Z), parent(Z, Y))

(A logic program)

More interesting ILP

Examples:
Some carcinogenic chemicals
Some non-carcinogenic chemicals
1000's

$+$

Background:
Molecular structure of chemicals
General chemical knowledge
10,000's

$\downarrow$

Hypothesis:
$\forall X$ carcinogenic(X) $\leftarrow \ldots$
$\ldots$
$\ldots$ 10's

Hypothesis formation and justification

Abduction. Process of hypothesis formation.

Justification. The degree of belief assigned to an hypothesis given a certain amount of evidence.

Logical setting for abduction

B	$= C_1 \wedge C_2 \wedge \ldots$	**Background**
E	$= E^+ \wedge E^-$	**Examples**
E^+	$= e_1 \wedge e_2 \wedge \ldots$	Positive Examples
E^-	$= \overline{f_1} \wedge \overline{f_2} \wedge \ldots$	Negative Examples
H	$= D_1 \wedge D_2 \wedge \ldots$	**Hypothesis**

Prior Satisfiability. $B \wedge E^- \not\models \Box$

Posterior Satisfiability. $B \wedge H \wedge E^- \not\models \Box$

Prior Necessity. $B \not\models E^+$

Posterior Sufficiency. $B \wedge H \models E^+$,

$B \wedge D_i \models e_1 \vee e_2 \vee \ldots$

More on this later

Probabilistic setting for justification

Bayes' Theorem

$$p(h|E) = \frac{p(h).p(E|h)}{p(E)}$$

Best hypothesis in a set $\mathcal{H}$ (ignoring ties)

$$H \;=\; \mathrm{argmax}_{h \in \mathcal{H}}\; p(h|E)$$

Learning Framework

Let X be a countable set of instances (encodings of all objects of interest) and D_X be a probability measure on X

Let $\mathcal{C} \subseteq 2^X$ be a countable set of concepts and $D_\mathcal{C}$ be a probablity measure on 2^X

Let $\mathcal{H}$ be a countable set of hypotheses and $D_\mathcal{H}$ be a probability measure (prior) over $\mathcal{H}$

Let the concept represented by $h \in \mathcal{H}$ be $c(h) \in \mathcal{C}$

Learning Framework (contd.)

Let $\mathcal{C}$ and $\mathcal{H}$ be such that

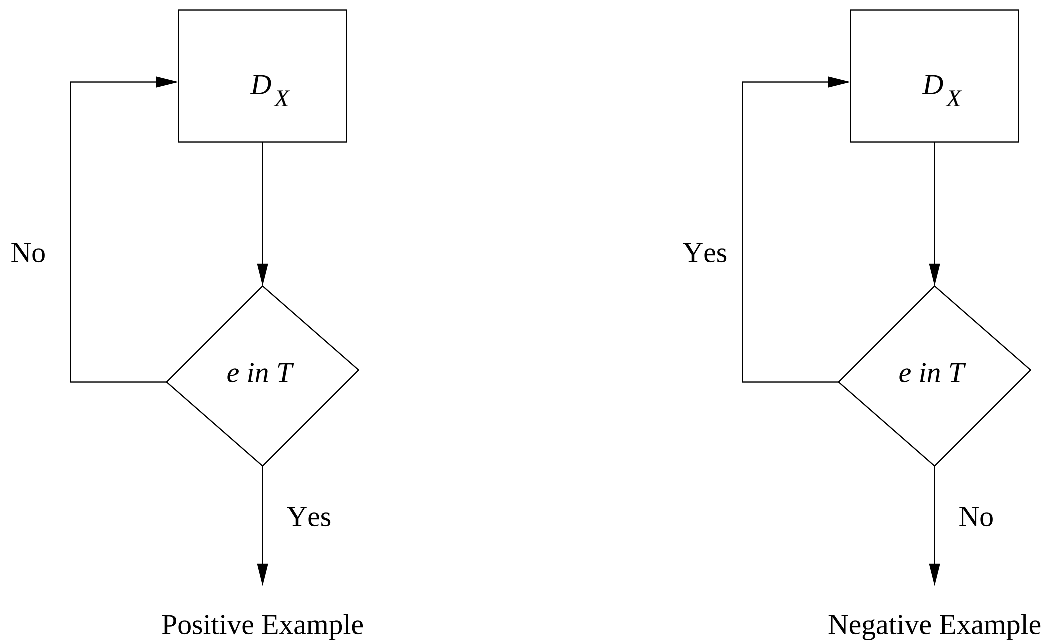
- for each $C \in \mathcal{C}$, there is an $h \in \mathcal{H}$ s.t. $C = c(h)$
- for each $C \in \mathcal{C}$, $D_{\mathcal{C}}(C) = \sum_{\{h \in \mathcal{H} | C = c(h)\}} P(h)$

Target concept T is chosen using the distribution $D_{\mathcal{C}}$

Let $g(h)$ denote the proportion (w.r.t. the instance space) of the concept represented by a hypothesis $h \in \mathcal{H}$

- That is, $g(h) = \sum_{x \in c(h)} D_X(x)$
- $g(h)$ is a measure of the "generality" of h

Model for Noise Free Data



Given $E = E^+ \cup E^-$

$$p(h|E) \propto D_{\mathcal{H}}(h) \prod_{e \in E^+} p(e|h) \prod_{e \in E^-} p(e|h)$$

Or

$$P(h|E) \propto D_{\mathcal{H}}(h) \prod_{e \in E^+} \frac{D_X(e)}{g(h)} \prod_{e \in E^-} \frac{D_X(e)}{1 - g(h)}$$

Noise Free Data (contd.)

Assuming p positive and n negative examples

$$P(h|E) \propto D_{\mathcal{H}}(h) \left(\prod_{e \in E} D_X(e)\right) \left(\frac{1}{g(h)}\right)^p \left(\frac{1}{1-g(h)}\right)^n$$

Maximal $P(h|E)$ means finding the hypothesis that maximises

$$\log D_{\mathcal{H}}(h) + p \ \log \frac{1}{g(h)} + n \ \log \frac{1}{1-g(h)}$$

If there are no negative examples, then this becomes

$$\log D_{\mathcal{H}}(h) + p \ \log \frac{1}{g(h)}$$

Some Questions

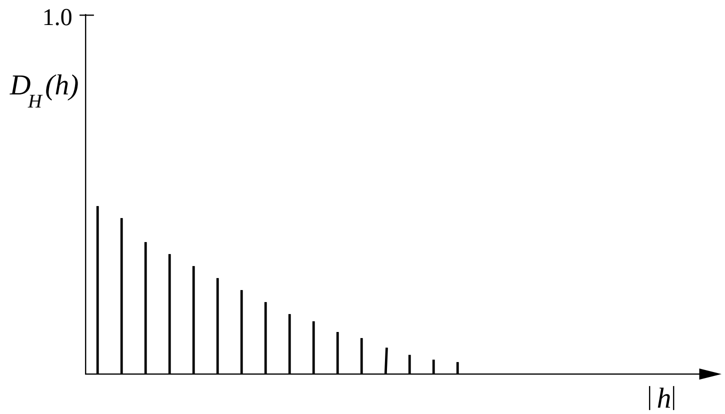
1. What is $D_{\mathcal{H}}(h)$?

2. What is $g(h)$?

3. What about noisy data?

The Distribution $D_{\mathcal{H}}$

A common assumption: "larger" programs are less likely (in coding terminology, require more bits to encode)



An example

$$D_{\mathcal{H}}(h) = 2^{-|h|}$$

That is

$$\log D_{\mathcal{H}}(h) = -|h|$$

The generality function g

Recall that $g(h) = \sum_{x \in c(h)} D_X(x)$

– $c(h)$ may be infinite

– D_X is usually unknown (and is a mapping to the reals)

Have to be satisfied with approximate estimates of $g(h)$

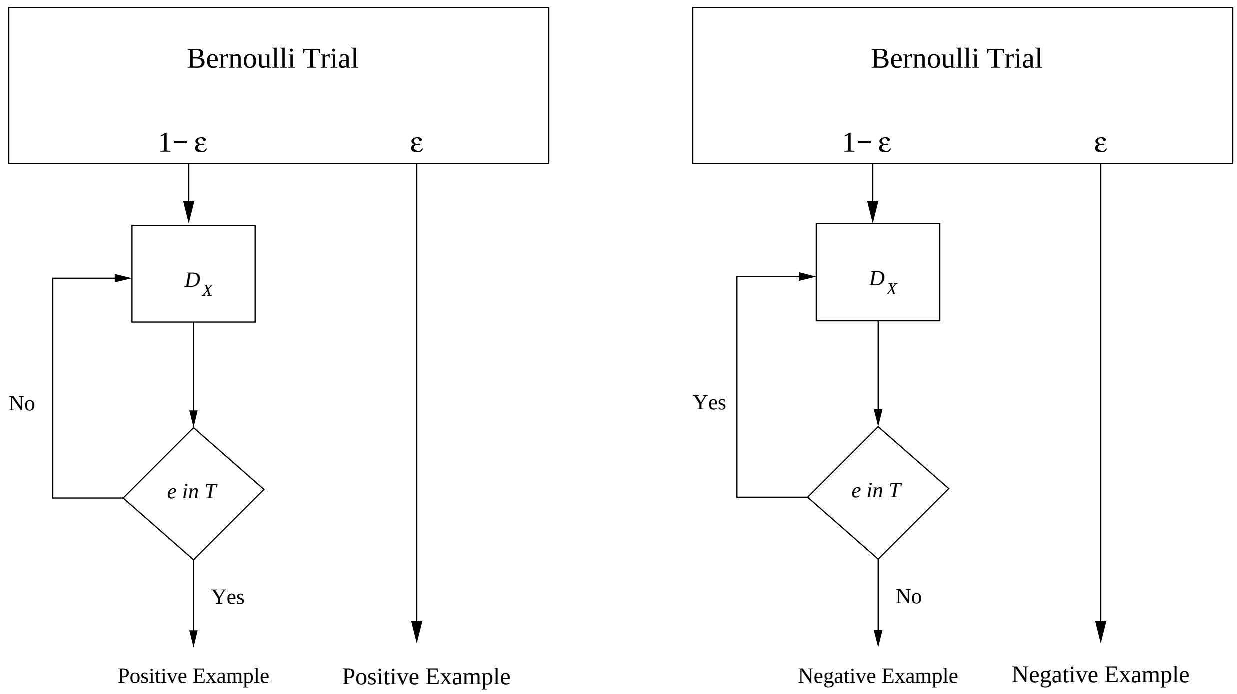
Estimation procedure

1. Randomly generate a finite sample of n instances using a known distribution (for eg. uniform)

2. Determine the number of these instances (say c) entailed by h

3. $g(h) \approx \frac{c+1}{n+2}$

A Model for Noisy Data



For any hypothesis h the examples $E = E^{+} \cup E^{-}$ can now be partitioned as follows

1. $TP = \{e | e \in E^{+}$ and $e \in c(h)\}$ (true positives)

2. $FN = \{e | e \in E^{+}$ and $e \notin c(h)\}$ (false negatives)

3. $FP = \{e | e \in E^{-}$ and $e \in c(h)\}$ (false positives)

4. $TN = \{e | e \in E^{-}$ and $e \notin c(h)\}$ (true negatives)

Noisy Data (contd.)

Recall

$$p(h|E) \propto D_{\mathcal{H}}(h) \prod_{e \in E^+} p(e|h) \prod_{e \in E^-} p(e|h)$$

Now

$$\prod_{e \in E^+} p(e|h) = \prod_{e \in TP} \left(\frac{D_X(e)(1-\epsilon)}{g(h)} + D_X(e)\epsilon \right) \prod_{e \in FN} D_X(e)\epsilon$$

$$\prod_{e \in E^-} p(e|h) = \prod_{e \in TN} \left(\frac{D_X(e)(1-\epsilon)}{1-g(h)} + D_X(e)\epsilon \right) \prod_{e \in FP} D_X(e)\epsilon$$

So, with $FPN = FP \cup FN$

$$p(h|E) \propto D_{\mathcal{H}}(h) \left(\prod_{e \in E} D_X(e) \right) \left(\frac{1-\epsilon}{g(h)} \right)^{|TP|} \left(\frac{1-\epsilon}{1-g(h)} \right)^{|TN|} \epsilon^{|FPN|}$$

Maximal $P(h|E)$ means finding the hypothesis that maximises

$$\log D_{\mathcal{H}}(h) + |TP| \log \frac{1-\epsilon}{g(h)} + |TN| \log \frac{1-\epsilon}{1-g(h)} + |FPN| \log \epsilon$$

Another Model for Noisy Data