# Inductive and Analytical Learning

<table>
<thead>
<tr>
<th>Inductive learning</th>
<th>Analytical learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hypothesis fits data</td>
<td>Hypothesis fits domain theory</td>
</tr>
<tr>
<td>Statistical inference</td>
<td>Deductive inference</td>
</tr>
<tr>
<td>Requires little prior knowledge</td>
<td>Learns from scarce data</td>
</tr>
<tr>
<td>Syntactic inductive bias</td>
<td>Bias is domain theory</td>
</tr>
</tbody>
</table>
What We Would Like

Inductive learning
Plentiful data
No prior knowledge

Analytical learning
Perfect prior knowledge
Scarce data

General purpose learning method:

• No domain theory $\rightarrow$ learn as well as inductive methods

• Perfect domain theory $\rightarrow$ learn as well as PROLOG-EBG

• Accomodate arbitrary and unknown errors in domain theory

• Accomodate arbitrary and unknown errors in training data
Domain theory:

\[
\text{Cup} \leftarrow \text{Stable, Liftable, OpenVessel}
\]

\[
\text{Stable} \leftarrow \text{BottomIsFlat}
\]

\[
\text{Liftable} \leftarrow \text{Graspable, Light}
\]

\[
\text{Graspable} \leftarrow \text{HasHandle}
\]

\[
\text{OpenVessel} \leftarrow \text{HasConcavity, ConcavityPointsUp}
\]

Training examples:

<table>
<thead>
<tr>
<th></th>
<th>Cups</th>
<th>Non-Cups</th>
</tr>
</thead>
<tbody>
<tr>
<td>BottomIsFlat</td>
<td>√ √ √ √</td>
<td>√ √ √ √</td>
</tr>
<tr>
<td>ConcavityPointsUp</td>
<td>√ √ √ √</td>
<td>√ √ √</td>
</tr>
<tr>
<td>Expensive</td>
<td>√ √</td>
<td>√ √ √</td>
</tr>
<tr>
<td>Fragile</td>
<td>√ √</td>
<td>√ √ √ √</td>
</tr>
<tr>
<td>HandleOnTop</td>
<td></td>
<td>√ √</td>
</tr>
<tr>
<td>HandleOnSide</td>
<td>√ √</td>
<td></td>
</tr>
<tr>
<td>HasConcavity</td>
<td>√ √ √ √</td>
<td>√ √ √ √ √ √</td>
</tr>
<tr>
<td>HasHandle</td>
<td>√ √</td>
<td>√ √ √</td>
</tr>
<tr>
<td>Light</td>
<td>√ √ √ √</td>
<td>√ √ √</td>
</tr>
<tr>
<td>MadeOfCeramic</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>MadeOfPaper</td>
<td></td>
<td>√ √</td>
</tr>
<tr>
<td>MadeOfStyrofoam</td>
<td>√ √</td>
<td></td>
</tr>
</tbody>
</table>
KBANN

KBANN (data $D$, domain theory $B$)
1. Create a feedforward network $h$ equivalent to $B$
2. Use BACKPROP to tune $h$ to fit $D$
Neural Net Equivalent to Domain Theory

Expensive
BottomIsFlat
MadeOfCeramic
MadeOfStyrofoam
MadeOfPaper
HasHandle
HandleOnTop
HandleOnSide
Light
HasConcavity
ConcavityPointsUp
Fragile

Stable
Graspable
Liftable
OpenVessel
Cup
Creating Network Equivalent to Domain Theory

Create one unit per horn clause rule (i.e., an AND unit)

- Connect unit inputs to corresponding clause antecedents
- For each non-negated antecedent, corresponding input weight $w \leftarrow W$, where $W$ is some constant
- For each negated antecedent, input weight $w \leftarrow -W$
- Threshold weight $w_0 \leftarrow -(n - .5)W$, where $n$ is number of non-negated antecedents

Finally, add many additional connections with near-zero weights

$$Liftable \leftarrow Graspable, \neg Heavy$$
Result of refining the network
KBANN Results

Classifying promoter regions in DNA leave one out testing:

- Backpropagation: error rate 8/106
- KBANN: 4/106

Similar improvements on other classification, control tasks.
Hypothesis space search in KBANN

Hypothesis Space

Initial hypothesis for KBANN

Hypotheses that fit training data equally well

Initial hypothesis for BACKPROPAGATION
EBNN

Key idea:

- Previously learned approximate domain theory
- Domain theory represented by collection of neural networks
- Learn target function as another neural network
Explanation of training example in terms of domain theory:

Target network:
Modified Objective for Gradient Descent

\[ E = \sum_{i} \left[ (f(x_i) - \hat{f}(x_i))^2 + \mu_i \sum_{j} \left( \frac{\partial A(x)}{\partial x_j} - \frac{\partial \hat{f}(x)}{\partial x_j} \right)^2 \right] \]

where

\[ \mu_i \equiv 1 - \frac{|A(x_i) - f(x_i)|}{c} \]

- \(f(x)\) is target function
- \(\hat{f}(x)\) is neural net approximation to \(f(x)\)
- \(A(x)\) is domain theory approximation to \(f(x)\)
Hypothesis Space Search in EBNN

Hypothesis Space

Hypotheses that maximize fit to data and prior knowledge

TANGENTPROP Search

Hypotheses that maximize fit to data

BACKPROPAGATION Search
Search in FOCL
FOCL Results

Recognizing legal chess endgame positions:

- 30 positive, 30 negative examples
- FOIL: 86%
- FOCL: 94% (using domain theory with 76% accuracy)

NYNEX telephone network diagnosis

- 500 training examples
- FOIL: 90%
- FOCL: 98% (using domain theory with 95% accuracy)