Combining Inductive and Analytical Learning

- Why combine inductive and analytical learning?
- KBANN: prior knowledge to initialize the hypothesis
- TangentProp, EBNN: prior knowledge alters search objective
- FOCL: prior knowledge alters search operators
### 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

- General purpose learning method:
- No domain theory $\rightarrow$ learn as well as inductive methods
- Perfect domain theory $\rightarrow$ learn as well as PROLOG-EBG
- Accommodate arbitrary and unknown errors in domain theory
- Accommodate arbitrary and unknown errors in training data

<table>
<thead>
<tr>
<th>Inductive learning</th>
<th>Analytical learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plentiful data</td>
<td>Scarce data</td>
</tr>
<tr>
<td>No prior knowledge</td>
<td>Perfect prior knowledge</td>
</tr>
</tbody>
</table>
Domain Theory

Cup ← Stable, Liftable, OpenVessel
Stable ← BottomIsFlat
Liftable ← Graspable, Light
Graspable ← HasHandle
OpenVessel ← HasConcavity, ConcavityPointsUp
## Training Examples

<table>
<thead>
<tr>
<th>Feature</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>MadeOfStyroForm</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
KBANN

Knowledge Based Artificial Neural Networks

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

large positive weight
large negative weight
negligible weight
Creating Network Equivalent to Domain Theory

Create one unit per horn clause rule (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, weight $w \leftarrow -W$
- Threshold weight $w_0 \leftarrow -(n - .5) W$, where $n$ is number of non-negated antecedents

Finally, add additional connections with near-zero weights

$Liftable \leftarrow Graspable, \neg Heavy$
Result of Refining the Network

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

Stable
Graspable
Liftable
OpenVessel

Cup

large positive weight
large negative weight
negligible weight
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

Hypotheses that fit training data equally well

Initial hypothesis for KBANN

Initial hypothesis for Backpropagation
EBNN

Explanation Based Neural Network

Key idea:
• Previously learned approximate domain theory
• Domain theory represented by collection of neural networks
• Learn target function as another neural network
Explanation in Terms of Domain Theory

Prior learned networks for useful concepts combined into a single target network

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

Stable
Graspable
Liftable
OpenVessel
Cup
TangetProp

Assume \(x, f(x)\) and \(\frac{\partial f(x)}{\partial x}\) provided as input

Modified objective for gradient descent:

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

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 TangentProp

Hypothesis Space

Hypotheses that maximize fit to data and prior knowledge

Hypotheses that maximize fit to data

TangentProp Search

Backpropagation Search
FOCL

• Adaptation of FOIL that uses domain theory
• When adding a literal to a rule, not only consider adding single terms, but also think about adding terms from domain theory
• May also prune specializations generated
Search in FOCL

```
Cup ← HasHandle [2+,3-]

Cup ← ¬HasHandle [2+,3-]

Cup ← Fragile [2+,4-]

Cup ← BottomIsFlat, Light, HasConcavity, ConcavityPointsUp [4+,2-]

Cup ← BottomIsFlat, Light, HasConcavity, ConcavityPointsUp, HandleOnTop [0+,2-]

Cup ← BottomIsFlat, Light, HasConcavity, ConcavityPointsUp, ¬HandleOnTop [4+,0-]

Cup ← BottomIsFlat, Light, HasConcavity, ConcavityPointsUp, HandleOnSide [2+,0-]
```
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)