Analytical Learning Discussion (4 of 4): Refinement of Approximate Domain Theories by Knowledge-Based Neural Networks

Friday, February 4, 2000
Lijun Wang
Department of Computing and Information Sciences, KSU

Presentation Outline
- **Paper**
  - "Refinement of Approximate Domain Theories by Knowledge-Based Neural Networks"
  - Authors: Geoffrey G. Towell, Jude W. Shavlik, Michiel O. Noordewier
  - Appears in the Proceedings of the Eighth National Conference on AI
- **Overview**
  - Use Horn clauses domain theory to create an equivalent artificial neural network (ANN)
  - KBANN algorithm
  - Empirical testing in molecular biology
  - Extension Research of KBANN
- **Application to Knowledge Discovery in Database: Issues**
  - Combined inductive and analytical learning
  - Key strengths: better than random initial weight? Lead to better generalization accuracy for the final hypothesis?
  - Key weakness: restricted to non-recursive, prepositional domain theories

KBANN Algorithm
- The KBANN assumes a domain theory can be represented by an ANN
  - **Definition of ANN**
    - An artificial neural network is composed of a number of nodes, or units, connected by links. Each link has a numeric weight associated with it. Learning takes place by updating the weights.
  - **Given**
    - A set of training examples
    - A domain theory consisting of nonrecursive, prepositional Horn clauses
  - **Determine**
    - An artificial neural network that fits the training examples, biased by the domain theory
    - the knowledge base is translated into ANN

KBANN Algorithm Examples
- An Illustrative Example (Translation of a Knowledge Base into an ANN)

KBANN Algorithm (continue)
- Translation of rules
  - sets weights on links and biases of units so that units have significant activation only when the corresponding deduction could be made using the domain knowledge
  - **Explanations**
    - for each mandatory antecedent, assign a weight: \( w \)
    - for each prohibitory antecedent, assign a weight: \(-w\)
    - bias on the unit: \( n \times w + \beta \) for conjunction \( w + \beta \) for disjunction
- **Algorithm specification**
  - overview
  - Translate rules to set initial network structure
  - Add units not specified by translation
  - Add links not specified by translation
  - Perturb the network by adding near zero random numbers to all link weights and biases

KBANN Algorithm (continue)
- Cup learning task (from Machine Learning by Tom Mitchell)
  - Domain theory
    - **Stable**, **Light**, **OpenVessel**
    - **Stable** --- **Bottleneck** --- **Stable**
    - **Light** --- **Graspable** --- **NonHandable**
    - **OpenVessel** --- **HasConcavity** --- **ConcavityPointsUp**
  - Neural Network
    - **Stable**, **Light**, **OpenVessel**
    - **Stable** --- **Bottleneck** --- **Stable**
    - **Light** --- **Graspable** --- **NonHandable**
    - **OpenVessel** --- **HasConcavity** --- **ConcavityPointsUp**
  - Training Examples
Experimenting with KBANN
• Molecular genetics experiment using KBANN
  - Task
    • learn to recognize DNA segments called promoter regions which influence gene activity
  - Domain theory
    • a promoter involves two subcategories: a contact and a conformation region
    • contact involves two regions
    • rules defining region characteristics
    • example:
      conformation:-@45 "axxx"
  - ANN

Experimenting with KBANN (continue)
• Molecular genetics problem using KBANN
  - procedure
    • 53 positive and 53 negative training examples
    • N = 106
    • “leave-one-out” method, on each iteration KBANN was trained using 105 of the 106 examples and tested on the remaining example
  - results

<table>
<thead>
<tr>
<th>System</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>KBANN</td>
<td>4/106</td>
</tr>
<tr>
<td>Standard Backpropagation</td>
<td>8/106</td>
</tr>
<tr>
<td>O’Neill</td>
<td>12/106</td>
</tr>
<tr>
<td>Nearest Neighbor</td>
<td>13/106</td>
</tr>
<tr>
<td>ID3</td>
<td>19/106</td>
</tr>
</tbody>
</table>

Related Work
• Problems specific to Neural Networks
  - Topology determination (restricted to a single layer of hidden units or random setting of link weights)
  - Integration of existing information into the network (how to use background information or improve incorrect domain theories in ANNs)

• KBANN solutions
  - Connect the inputs of network units to the attributes tested by the clause antecedents, assign a weight of w to the unit for each positive antecedent or -w for each negative antecedent
  - Initialize the hypothesis to perfectly fit the domain theory, then inductively refine the initial hypothesis as needed to fit the training data

Summary Points
• Content Critique
  - Key contribution:
    • Analytically creates a network equivalent to the given domain theory
    • Inductively refines the initial hypothesis to better fit the training data
  - In doing so, it modifies the network weights to overcome the inconsistencies between the domain theory and the observed data.

  - Strengths
    • Generalize more accurately given an approximately correct domain theory
    • Outperform other purely inductive methods when data is scarce
    • Domain theory used in KBANN indicates important features to an example classification
    • Derived features are also specified through deduction, therefore reducing the complexity of an ANN’s final decision

  - Weaknesses
    • Is restricted to non-recursive, propositional (i.e., variable-free) Horn clauses
    • May be misled given highly inaccurate domain theory
    • Is problematic to extract information from ANNs after learning because some weight settings have no direct Horn clause analog.
    • Blackbox method, which provide good results without explanation

Summary Points (continue)
• Presentation Critique
  - Audience: AI (learning, planning), ANN, applied logic researchers
  - Positive and exemplary points
    • Clear example illustrating the translation of knowledge base into an ANN
    • Good experimental results over other inductive learning algorithm
  - Negative points and possible improvements
    • We understand some basic ideas of ANN translation, but still may not be able to do it

Questions, Comments