Lecture 11

Artificial Neural Networks (1 of 4)

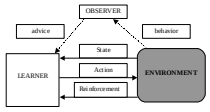
Friday, February 11, 2000

Kiran Nandivada
Department of Computing and Information Sciences, KSU

Readings:
"Incorporating Advice into Agents that Learn from Reinforcements"
Richard Maclin and Jude W. Shavlik

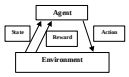
Presentation Overview

- Paper
  - "Incorporating Advice into Agents that Learn from Reinforcements"
  - Authors: Richard Maclin and Jude W. Shavlik, Computer Sciences Department, University of Wisconsin
- Overview
  - Learning from reinforcements by accepting advice from an external observer
- Goals
  - The system accepts the advice
  - The external observer can provide advice at any time



Terminology

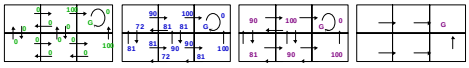
- Reinforcement learning
  - Reward or Reinforcement
    - Feedback provided to the agent for the action it performed in the previous state
  - Task of learning
    - The agent learns from this reward and chooses actions that produce highest cumulative reward (Mitchell, Ch. 13)



- Given
  - Observation sequence $s_0 \xrightarrow{a_0;r_0} s_1 \xrightarrow{a_1;r_1} s_2 \xrightarrow{a_2;r_2} \cdots$
  - Discount factor $\gamma \in [0, 1)$
- Learn to: choose actions that maximize $r(t) + \gamma r(t+1) + \gamma^2 r(t+2) + \ldots$

Terminology

- Q-learning
  - The agent learns a numerical evaluation function defined over states of actions, and then implement an optimal policy in terms of this evaluation function (Mitchell, Ch. 13)



- Connectionist Q-learning
  - The utility function is implemented as neural network, whose inputs describe the current state and whose outputs are the utility of each action

Presentation Outline

Issues
- Is the advice given by the external observer used effectively
- Does it matter in this type of learning 'when' the advice is given
- Key strengths - the use of external observer enhanced the learning process
- Key weaknesses - accepts only single advice at a time

Outline
- Advice taking
  - Proposed a strategy where several steps described by Hayes-Roth, Klahr, and \ Mostow(1981), can be achieved using reinforcement learning.
- Experiments
- Test Environment
- Results
- Future work
- Summary

Advice-taking

- Step 1 - Provide advice to the agent
  - Advice is provided by the external observer whenever the observer feels appropriate
- Step 2 - Convert the advice to an internal representation
  - Expression of advice is in the form of a simple programming language and list of terms which specifies certain tasks.
- Step 3 - Convert the advice into an usable form
  - Operationalize the advice - conversion of advice into interpretable statements
  - Requires a compiler for certain task specific terms
- Step 4 - Integrate the reformulated advice into the agent's current knowledge base
  - Used an extended KBANN approach
  - Rules are installed "incrementally" into the network
  - Insert advice in to the network (connectionist representation of the utility function) at any time during learning
  - Example - Agent learning to play a video game
  - A sample version of the advice provided to the agent

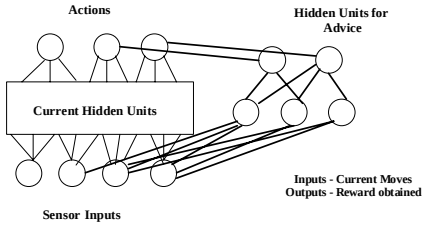
Advice-taking

- A sample version of advice
- Advice

```
IF  An Enemy        IS    (Near  West)
    An Obstacle     IS    (Near  North)
THEN
   MULTIACTION
     MoveEast  MoveNorth
     END;
WHEN      Surrounded
          OKtoPushEast
          An Enemy IS Near
REPEAT
          PushEast
          MoveEast
UNTIL     ¬ OKtoPushEast
          ¬ Surrounded
```

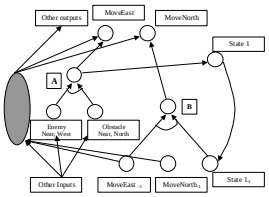
Advice-taking

- Network showing the advice added by adding hidden units that correspond to the advice



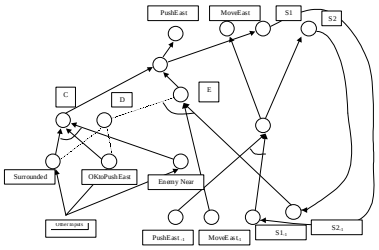
Advice-taking

– Allows advice that contains multi-step plans



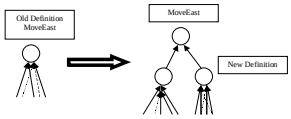
Advice-taking

– Allow advice that contains loops



Advice-taking

– Allow advice that refers to previously defined terms



Experimentation

– Step 5 - Judging the value of advice
  - Introduces a Q-learning concept to "wash-out" a poor advice
  - Empirically evaluate the new advice
  - Retracts or counteract a bad advice

– Experiments
  - Goal - Empirically evaluate whether this particular approach of providing advice is better
  - Hypothesis 1 - System takes advantage of the advice
  - Hypothesis 2 - Observer provides appropriate advice to the agent at any time during the training

– Test Environment
  - Agent performs certain actions which include *moving* and *pushing* in the directions East, West, North, and South and *doing nothing*

Experimentation

- Assumes an agent-centered model partitioned into sectors
- Agent calculates the percentage (input to the network) of the type of the object occupied in each sector

– Methodology
  - Train the agents for a fixed number of episodes
  - Choose an initial topology
  - Provide advice to the agent
  - Four forms of advice are provided to the agent (ElimEnemies, Surrounded, SimpleMoves, NonLocalMoves)

– Result

Experiment 1
  - Train the agent initially
  - Measure the value of adding advice
  - Add the advice and measure the test set reinforcement

Results

Advice added	Avg Total Reinforcement
Baseline (none)	1.32
SimpleMoves	1.92
NonLocalMoves	2.01
ElimEnemies	1.87
Surrounded	1.72

Exp 1 - Average total reinforcement for the four sets of advice (significant with 99% confidence)

– Significance
  - Reported the gains obtained over the baseline or without the addition of advice
  - Initial training was for 1000 episodes and system training after adding advice was for 2000 episodes and the baseline for 3000 episodes

Experiment 2
  - Insert advice at different times during the training (0,1000, and 2000 episodes)
  - Convergence to same amount of reinforcement irrespective of the time the advice was provided
  - Observe whether each task corresponding to each piece of advice is performed

Results

Advice added	Enemies	Rewards	Survival time
None	0.15	3.09	32.7
SimpleMoves	0.28	3.79	39.6
NonLocalMoves	0.26	3.95	39.1
ElimEnemies	0.44	3.50	38.3
Surrounded	0.30	3.48	46.2

Exp 2 - Mean no. of enemies captured, rewards collected, and number of actions taken for the experiments

– Significance
  - Suggests the effect of advice and subsequent refinement of the advice by the system
  - Graphs in the paper indicate how the training improved the convergence of the reinforcements after certain number of episodes

Summary

- Future Work
  – Accepting multiple pieces of advice at different times during training
  – Evaluate periodic re training or "replay" on certain pairs of states
- Possible improvements
  – a) Temporal difference method b) Also,can improve on their advice taking strategy and use EBL to improve on pieces of advice and use it to solve any advice-taking problems
- Content Critique
  – Key Contribution - A good example of how the learner can accept "general advice" at any time during the training (agent learning naturally)
  – Strengths - Proved that the advice improves the expected rewards
  – Weaknesses - No reference to convergence in terms of error and generalization
- Presentation Critique
  – Audience - AI, Robotics (principally concerns game playing strategically)
  – Positive points - Good introduction and explanation regarding advice taking steps
  – Negative points - Results - Did not mention what kind of statistical methods were used to get the significance they reported.