Reinforcement Learning

Jacky Baltes

University of Auckland

Email: j.baltes@auckland.ac.nz

Web: www.tcs.auckland.ac.nz/~jacky

March 7, 2003

Reinforcement Learning in Robotics

- Unsupervised learning technique. Also GA/GP, ANN.
- Paradigm: Agent acts in the world and at some time receives feedback (**reward**).
- Chess: one reward at the end of the game (**terminal state**).
- Ping pong: each point
- Controller: every move?
- Reward is almost always delayed.
- Problem: Credit assignment. Which move was to blame?

Reinforcement Learning Paradigm

- In/accessible Environment
- Passive/active learner
- Learning of state utility functions
- Learning of action values (**Q-learning**)

Passive Learning

- Observe state transitions
- Reward is additive, fixed reward for a state
- Compute reward for current state: **reward-to-go**
- LMS Update: running average for state utilities
- Utility of a state is constrained by its successors and their probabilities.
- $U(i) = R(i) + \sum_j M(i,j)U(j)$
- Adaptive Dynamic Programming (ADP)
- Only need the value for states we actually visit
- TD learning:
  $U(i) = U(i) + \alpha(N(i))(R(i) + U(j) - U(i))$

Active Learning

- Agent has a choice to make (action a)
- $U(i) = R(i) + \max(a) \sum_j M(a,i,j)U(j)$
- Update environment model ($M(i,j)$) with action (a).
- TD learning unchanged:
  $U(i) = U(i) + \alpha(N(a,i))(R(i) + U(j) - U(i))$
- Q-learning (state-action value)
- $U(i) = \max(a)Q(a,i)$
- Q-update
- $Q(a,i) =$
  $Q(a,i) + \alpha(R(i) + \max(a_j)Q(a_j,j) - Q(a,i))$
- Don't need a state model

Problems

- Exploration vs Exploitation?
- Choose the best action $\Rightarrow$ premature convergence
- Suboptimal solutions
- Exploration function: Assign optimistic estimate to unexplored states (e.g., 1000 if visited less than 10 times)
- Explicit and implicit representation of $Q(a, i)$.
- Generalization over states
- Function approximation:
  - Inductive Learning ID3
  - ANN
  - Case-based function
- State aliasing