

Reinforcement learning and action-state values

David Keil
University of Connecticut, October 2002

Sources: S. Russell and P. Norvig, AI: A modern approach (Prentice Hall, 1995); R. Sutton and A. Barto, Reinforcement learning: An introduction (MIT, 1998)

Learning agents

- In learning, percepts improve agent's future success in interaction
- Components:
 - Learning element (improves policy)
 - Performance element (executes policy)
 - Critic: Applies fixed performance measure
 - Problem generator: Suggests experimental actions that will provide information to learning element

RN95, pp. 525ff

Passive ADP learning

- Adaptive dynamic programming
- Requires transition model, containing information about structure of environment
- Utility U of state i is computed from state's reward R and utilities of all adjacent states, weighted by transition probabilities:

$$U(i) = R(i) + \sum_j M_{ij}U(j)$$

RN95, p. 603

Temporal difference (TD) learning

- Uses observed transitions and differences between utilities of successive states to adjust utility estimates
- Update rule based on transition from state i to j :

$$U(i) \leftarrow U(i) + \alpha(R(i) + U(j) - U(i))$$
 where
 - U is utility,
 - R is reward
 - α is learning rate

RN95, p. 604

Model-based vs. model-free learning

- Transition model is used in adaptive dynamic programming (ADP) learners
- Use of models is associated with knowledge-based AI
- Temporal difference (TD) learning does not use model
- Q learning is model free

RN95, pp. 612-615

Q-values

- Definition: Q-values are values $Q(a, i)$ of expected utility associated with a given action in a given state
- Utility of state:

$$U(i) = \max_a Q(a, i)$$
- Q-values permit decision making without a transition model
- Q-values are directly learnable from reward percepts

TDL-based Q-learning agent

Agent attributes:

Q = table of (action, state) values
 N = number of previous (action, state) visits
 $R[s]$ = reward for state s
 a = previous action
 s = previous state s' = current state
 α = learning rate f = exploratory function

Algorithm:

$N[a, s] \leftarrow N[a, s] + 1$
 $Q[a, s] \leftarrow Q[a, s] + \alpha(R[s] + \max_{a'} Q[a', s'] - Q[a, s])$
 $s \leftarrow s'$
return $\arg \max_{a'} f(Q[a', s], N[a', s'])$

RN95, p. 614

Explicit vs. implicit representation

- Learned function may be represented as table (explicit)
- Policies for games with 10^{120} states have been represented as weights in a linear function of board features (implicit)
- Implicit representation allows generalization from states visited to those not

RN95, pp. 615-616

Value function methods vs. evolutionary methods of RL

- Evolutionary methods evaluated a policy that is fixed over many test sequences
- Value function methods update policy during execution
- Value function update benefits from information obtained during interaction

SB98, p. 13