MAXQ HRL in Soar

Mitchell Keith Bloch

University of Michigan

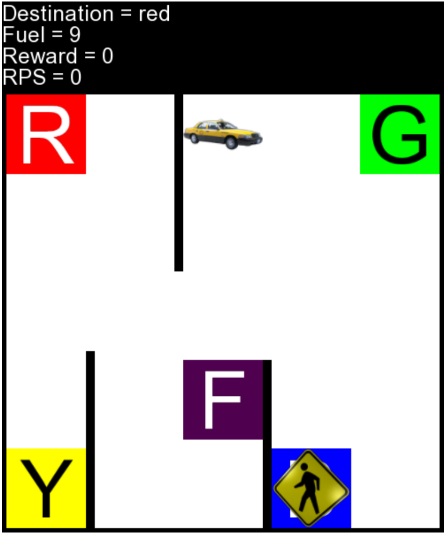
May 17, 2010

Motivation

1. Replicate the results described in [Dietterich, 1998]
2. Determine how to bring the cooling techniques employed by a special purpose one-off technique (MAXQ) to a general purpose architecture (Soar)
3. Demonstrate advantages of MAXQ HRL over flat RL
4. Demonstrate value of MAXQ HRL cooling techniques

Outline

Basic Information



1. Initial conditions
   a. 5x5 grid world
   b. 4 sources/destinations
   c. Refueling station
   d. Impassable walls
   e. [5,12] fuel, capped at 14
2. Goals
   a. Pick up passenger
   b. Deliver to destination
   c. Avoid running out of fuel
   d. Always achievable
3. Rewards
   a. -1 for a legal action
   b. -10 for an illegal action
   c. -20 for running out of fuel
   d. 20 for delivering the passenger

Reinforcement Learning

1. Reinforcement learning problem
   a. Agent
   b. Environment and reward signal
2. Q-learning–a temporal difference (TD) method
3. TD methods involve a value function
   a. Expected future reward
   b. One value per action per state in the limit
4. Should converge on optimal policy
   a. Learn value function
   b. Stop exploring

MAXQ Hierarchical Reinforcement Learning

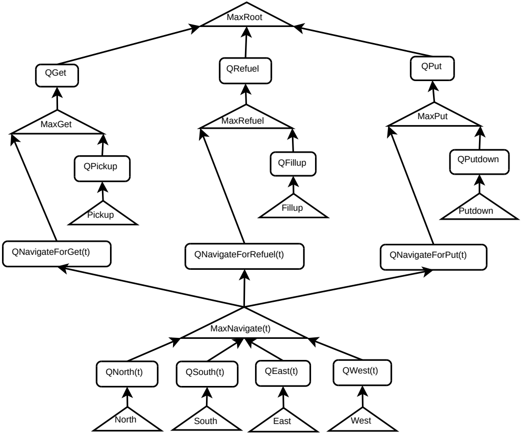


Figure: Dietterich's MAXQ Hierarchy.

1. Formulated by [Dietterich, 1998]
2. Max nodes represent goals
   a. Each goal is an RL problem
   b. Each has its own cooling strategy
3. A Max node cools on success if the absolute Bellman error per step is low
   a. Assumes success
   b. Assumes deterministic environment

Soar-RL

```
sp {reinforce*putdown*151
    (state <s> ^operator <o> +)
    (<o> ^name putdown
         ^passenger true
         ^x 0 ^y 0
         ^destination yellow)
  -->
    (<s> ^operator <o> = 20.)
}
```

Figure: Abstract view of a putdown proposal

1. Proposal rules assigned Q values
2. Boltzmann indifferent-selection decides between proposals
3. Q values modified when rewards received

Boltzmann indifferent-selection

$$\frac{e^{\frac{Q(s,O_i)}{\tau}}}{\sum_{j=1}^{n} e^{\frac{Q(s,O_j)}{\tau}}}$$

Figure: Boltzmann indifferent-selection prefers actions with higher Q values

1. Start with a high temperature
   a. Choose almost randomly
2. End with a low temperature
   a. Choose the best almost exclusively
3. Interpolate in between

Outline

Architectural Modifications

Cooling schedule for HRL proposed and implemented by [Dietterich, 1998]

1. Support per-goal cooling schedules
2. Slow cooling
   a. Require low average absolute Bellman error per step
   b. Require success

Outline

Basic Details

1. Agent knows
   a. Taxi's position
   b. Current type of cell
   c. Fuel available
   d. Where the passenger is
   e. Where the passenger wants to go
2. Seven choices of action from any state
3. Environment provides rewards

Flat RL Agent

1. Actions unrestricted
2. Pickup and Putdown coded coarsely

Basics

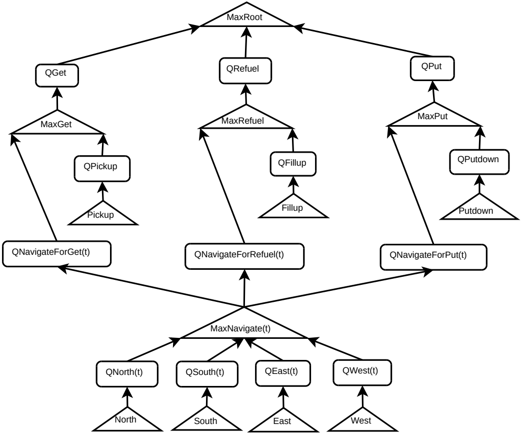


Figure: Dietterich's MAXQ Hierarchy.

1. Max nodes represent plans
2. Q values represent knowledge of implementation
3. Much more coarse coding

Reward Assignment

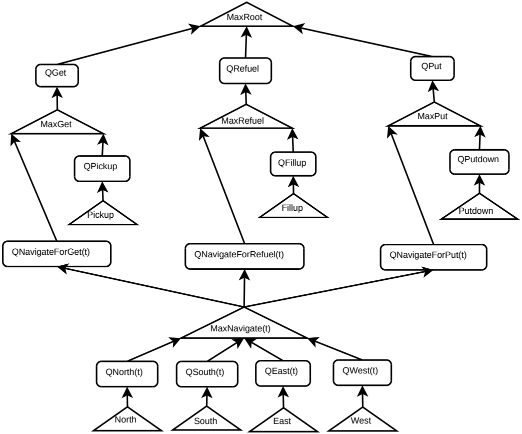


Figure: Dietterich's MAXQ Hierarchy.

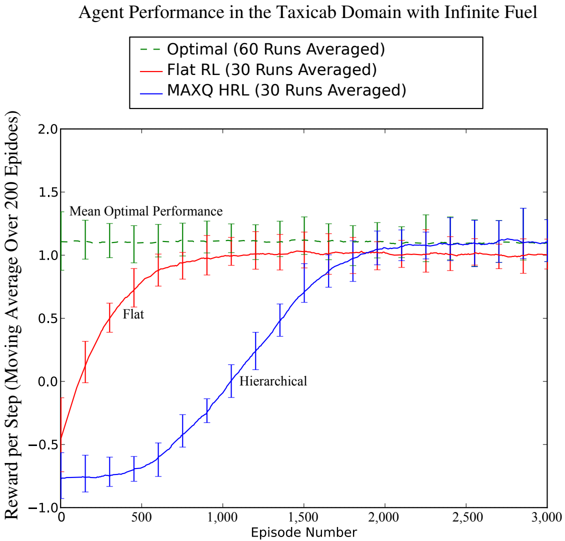
1. ± 20 passed to MaxRoot
2. -10 passed to MaxGet, MaxPut, and MaxRefuel
3. -1 passed to all layers of the hierarchy
4. Internal reward of 10 generated for the MaxGet
5. Internal reward of 10 generated for the MaxRefuel

Outline

Motivation and Overview

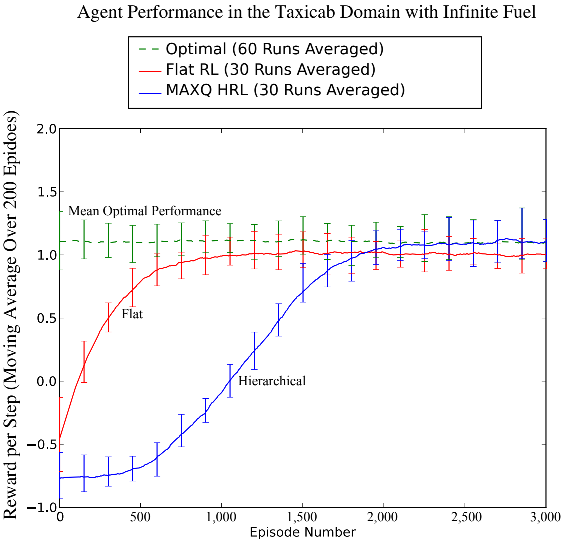
1. Wanted to replicate the result from [Dietterich, 1998], that MAXQ hierarchical reinforcement learning is superior to flat reinforcement learning in a task as difficult as the taxicab domain
2. Wanted to show that the cooling schedule of MAXQ offers an advantage over HRL without MAXQ
3. In both the finite task and the infinite task, the non-MAXQ HRL agent was changed in the following ways:
   a. Only one temperature for the whole agent
   b. Absolute Bellman error per step is ignored
   c. Failure is ignored for purposes of disabling learning and cooling

Plot Information



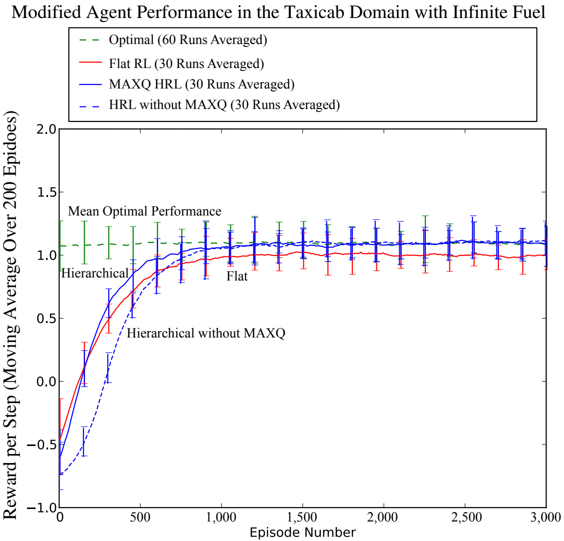
1. Plots are averaged over 30 sets of episodes
2. Afterward, they are smoothed using a moving average with a window of 200 episodes
3. Error bars indicate minima and maxima

Flat vs MAXQ HRL



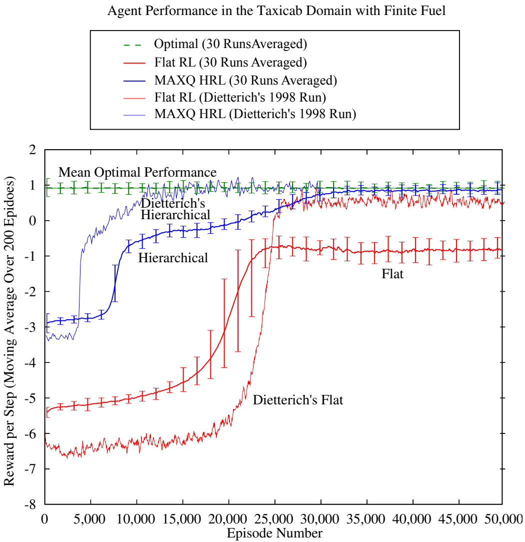
1. The HRL agent was untuned, and used the same parameters as the agent for the finite fuel task
2. After disabling exploration after $3,000$ episodes
   a. The optimal reward possible over $5,000$ episodes was 1.09 reward per step
   b. The flat agent averaged 1.00 reward per step
   c. The hierarchical agent averaged 1.09 reward per step and matched the optimal for all $5,000$ episodes in all 30 runs

Effect of MAXQ



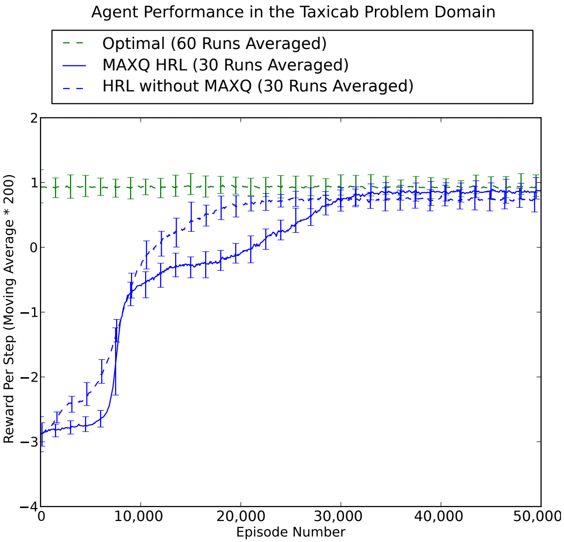
1. Results from the same HRL agent with all cooling rates reduced to 0.97 are plotted against the previous flat agent results
2. This new choice of cooling rate for all Max nodes was untuned
3. The hierarchical agent still averaged 1.09 reward per step but only matched the optimal for all $5,000$ episodes in 28 runs this time
4. Without Dietterich's cooling techniques, learning slowed significantly, but the agent averaged 1.10 reward per step and matched the optimal for all $5,000$ episodes in all 30 runs

Flat vs MAXQ HRL



1. After disabling exploration after 50, 000 episodes
   a. The optimal reward possible over 5, 000 episodes was 0.93 reward per step
   b. The flat agent averaged −0.83 reward per step and the hierarchical agent averaged 0.86 reward per step
2. My hierarchical Soar agent learns more slowly than that of [Dietterich, 1998], although both manage to achieve a virtually optimal policy by the end of 50, 000 runs

Effect of MAXQ



1. Once Dietterich's cooling techniques are disabled, learning actually speeds up a bit
2. However this agent averaged only 0.75 reward per step, which is significantly less than the 0.86 received when using these techniques

Outline

Future Work - Cooling Strategies

1. Important to take from [Dietterich, 1998] the importance of cooling on success
2. Exploration of more finely grained cooling strategies
3. Possible features of a successful strategy
   a. Pass back success signal with rewards
   b. Keep track of moving average of success rate
   c. Map this average to a temperature
   d. Use maximum temperature of available options
4. Possible goals
   a. Better fit temperature to learning
   b. Make coarse coding more useful

Outline

Mineral Resources

Nuggets

1. Implementing the cooling strategies employed by [Dietterich, 1998] in Soar was straightforward
2. The cooling strategies of MAXQ HRL have been integrated into Soar
3. The value of MAXQ HRL over flat RL has been verified
4. Shown that the MAXQ cooling strategies are of value

Coal

1. Need to be able to evaluate success
2. Unclear that the problem formulation is identical to [Dietterich, 1998]
3. Unable to reproduce Dietterich's level of success with the flat RL agent
4. No public release of architectural modifications yet

Thomas G. Dietterich.
The maxq method for hierarchical reinforcement learning.
In *In Proceedings of the Fifteenth International Conference on Machine Learning*, pages 118–126. Morgan Kaufmann, 1998.