1 Search

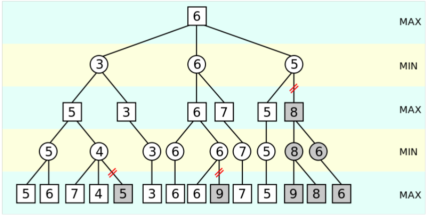
2 CSP

3 Adversarial Search

3.1 Minimax

Maximize your minimum gain. Assume opponent plays optimally.

3.1.1 Alpha-Beta Pruning



Idea: If we are at a max-node, and we found the value v of a child. The value of the max-node must be $\geq v$. Leverage this information when searching the other children.

3.2 Expectimax

Minimize the expected value of your opponents actions. Assume opponents play randomly instead of optimally. Can prune if we have bounds on the utility of terminal states. With this info, we can compute upper and lower bounds on the expected value of a chance node.

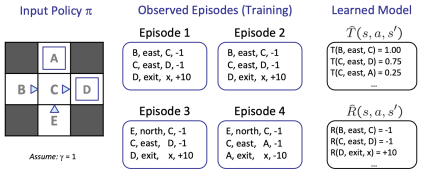
4 Utilities

5 MDPs

6 RL

Idea: Model world as MDP, but we don't know T and R.

Model-Based RL Approximate T and R based on experiences. Average R and T across epsiodes.



Assume: $\gamma = 1$

Observed Episodes (Training)

Episode 1

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 2

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 3

E, north, C, -1
C, east, D, -1
D, exit, x, +10

Episode 4

E, north, C, -1
C, east, A, -1
A, exit, x, -10

Learned Model

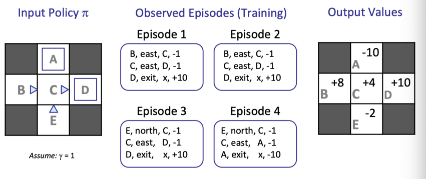
$\hat{T}(s, a, s')$

T(B, east, C) = 1.00
T(C, east, D) = 0.75
T(C, east, A) = 0.25
...

$\hat{R}(s, a, s')$

R(B, east, C) = -1
R(C, east, D) = -1
R(D, exit, x) = +10
...

Model-Free RL Direct Evaluation: Learn values of states from episodes.



Assume: $\gamma = 1$

Observed Episodes (Training)

Episode 1

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 2

B, east, C, -1
C, east, D, -1
D, exit, x, +10

Episode 3

E, north, C, -1
C, east, D, -1
D, exit, x, +10

Episode 4

E, north, C, -1
C, east, A, -1
A, exit, x, -10

$V(A) = -10/1 = -10$ $V(B) = (10*3)/3 = 10$ $V(C) = (\sum R + V(next))/4 = (-1*4+10*3+-10*1)/4 = 4$

TD-Learning: Fix π, learn V of states from episodes.

7 Bayes Net

8 HMMs

9 Particle Filters

10 Decision Networks

11 Naive Bayes

12 Perceptrons

13 Clustering

14 Deep Learning