Lecture 18 (Better Computations for Value Function)

Water flowing from goal states to the other states is a good analogy to visualise value iteration.

1 Asynchronous Dynamic Programming

1. Back up states individually
2. It is guaranteed to converge if all states continue to be selected

1.1 In-Place Dynamic Programming

1. Synchronous value iteration stored two copies of value function, the old and new state
2. In-place iteration only stores a single copy of the value funciton
3. Saves space (and convergence is quicker?)

1.2 Prioritised Sweeping

1. Use magnitude of Bellman error (change in the value of $v(s)$) to guide state selection
2. Backup the state with the largest error
3. Use a priority queue and add neighbours to queue after update

2 Problems with Value Iteration

The optimal policy converges much before the values converge.

2.1 Policy Iteration

1. Calculate the utilities for some fixed policy until convergence
2. Update policy using one step look ahead with resulting converged utilities as future values
3. Repeat the above until the policy converges
4. Can directly solve the Bellman equation in $O(s^3)$
5. Alternatively, we can also perform a few iterations to get an idea about the value function instead of the exact value