Reinforcement Learning (INF11010)

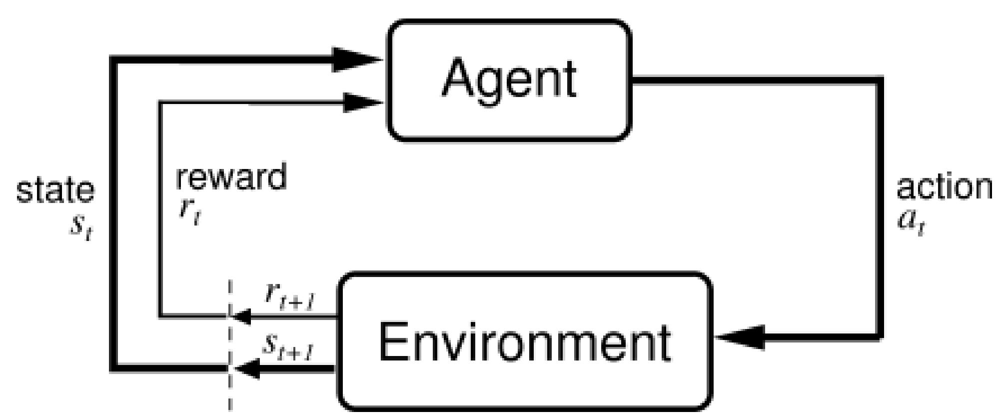
Lecture 5: Dynamic Programming for Reinforcement Learning

Pavlos Andreadis, January 30th 2018

Markov Decision Processes

- A finite Markov Decision Process (MDP) is a tuple (S, A, P, R, γ) where:

- S is a finite set of states
- A is a finite set of actions
- P is a state transition probability function
- R is a reward function
- γ is a discount factor



Today’s and Friday’s Content

- Dynamic Programming (DP) solutions to the RL problem
- Policy Evaluation + Policy Improvement →

  Policy Iteration || Value Iteration
- Backup diagrams and the Bellman Equation
- Generalised Policy Iteration
- Asynchronous Dynamic Programming
- Dynamic Programming methods in relation to other approaches

Dynamic Programming

- Algorithms for optimal policies given a *perfect model* of the environment as a Markov decision process (MDP)
- … but of theoretical importance.

- Applicable for exact solutions with discrete state & action model:

$$P^a_{s,s'} = Pr\{s_{t+1} = s' | s_t = s, a_t = a\}$$

$$R^a_{s,s'} = E\{r_{t+1} | a_t = a, s_t = s, s_{t+1} = s'\}$$

- … and provide approximate solutions for continuous problems.

Bellman Optimality Equations

$$
\begin{aligned}
V^*(s) &= max_a E\{r_{t+1} + \gamma\, V^*(s_{t+1}) | s_t = s, a_t = a\} \\
&= max_a \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma\, V^*(s') \right]
\end{aligned}
$$

$$
\begin{aligned}
Q^*(s,a) &= E\Big\{ r_{t+1} + \gamma\, max_{a'} Q^*(s_{t+1}, a') | s_t = s, a_t = a \Big\} \\
&= \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma\, max_{a'}\, Q^*(s', a') \right]
\end{aligned}
$$

for all $s \in S, a \in A(s),$ and $s' \in S.$

Policy Evaluation

- There exists a unique solution as long as $\gamma < 1$ or termination is guaranteed:

$$
\begin{aligned}
V^\pi(s) &= E_\pi\{r_{t+1} + \gamma\, r_{t+2} + \gamma^2\, r_{t+3} + ...|s_t = s\} \\
&= E_\pi\{r_{t+1} + \gamma V^\pi(s_{t+1})|s_t = s\} \\
&= \sum_a \pi(s,a) \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma V^\pi(s')\right]
\end{aligned}
$$

- … which is a system of |S| linear equations with |S| unknowns

Iterative Policy Evaluation

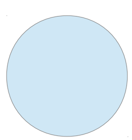
- An iterative solution, starting from an arbitrary V_0 (but with terminal states having a value of 0) and computing…

$$
\begin{aligned}
V_{k+1}(s) &= E_\pi\{r_{t+1} + \gamma V_k(s_{t+1}) | s_t = s\} \\
&= \sum_a \pi(s,a) \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma V_k(s') \right]
\end{aligned}
$$

- … which converges to V^π as $k \to \infty$
- At every iteration, every state is *backed up*
- For DP, this is a *full backup*, since we don't sample next states

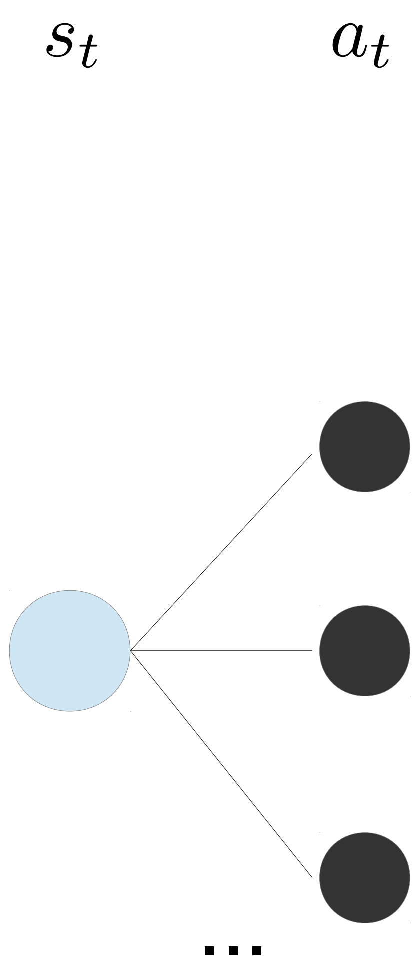
Backup Diagrams

s_t



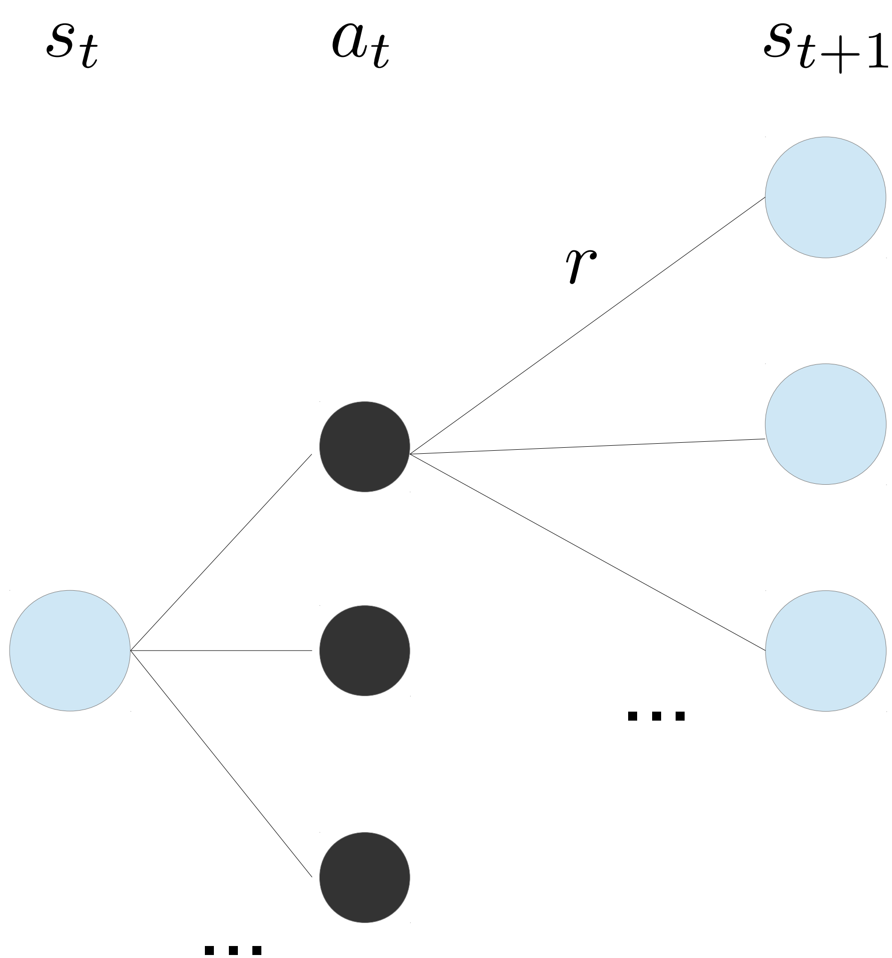
- State value function V

Backup Diagrams



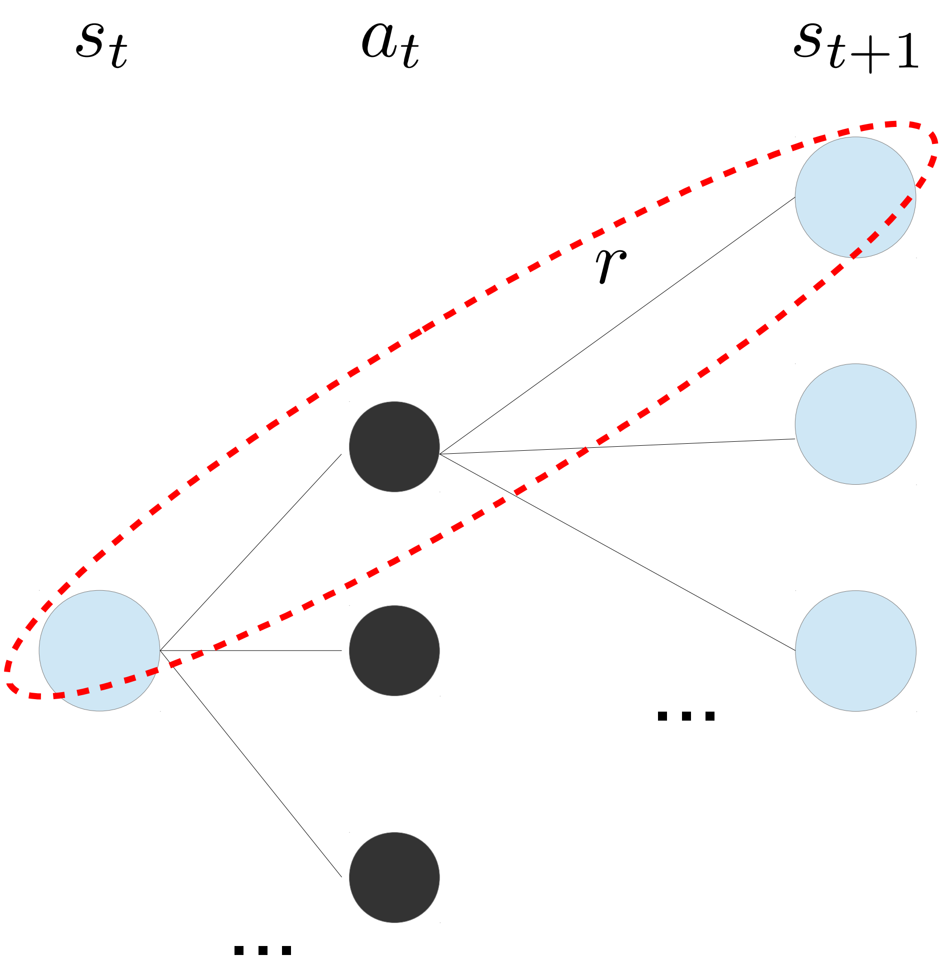
- State value function V

Backup Diagrams



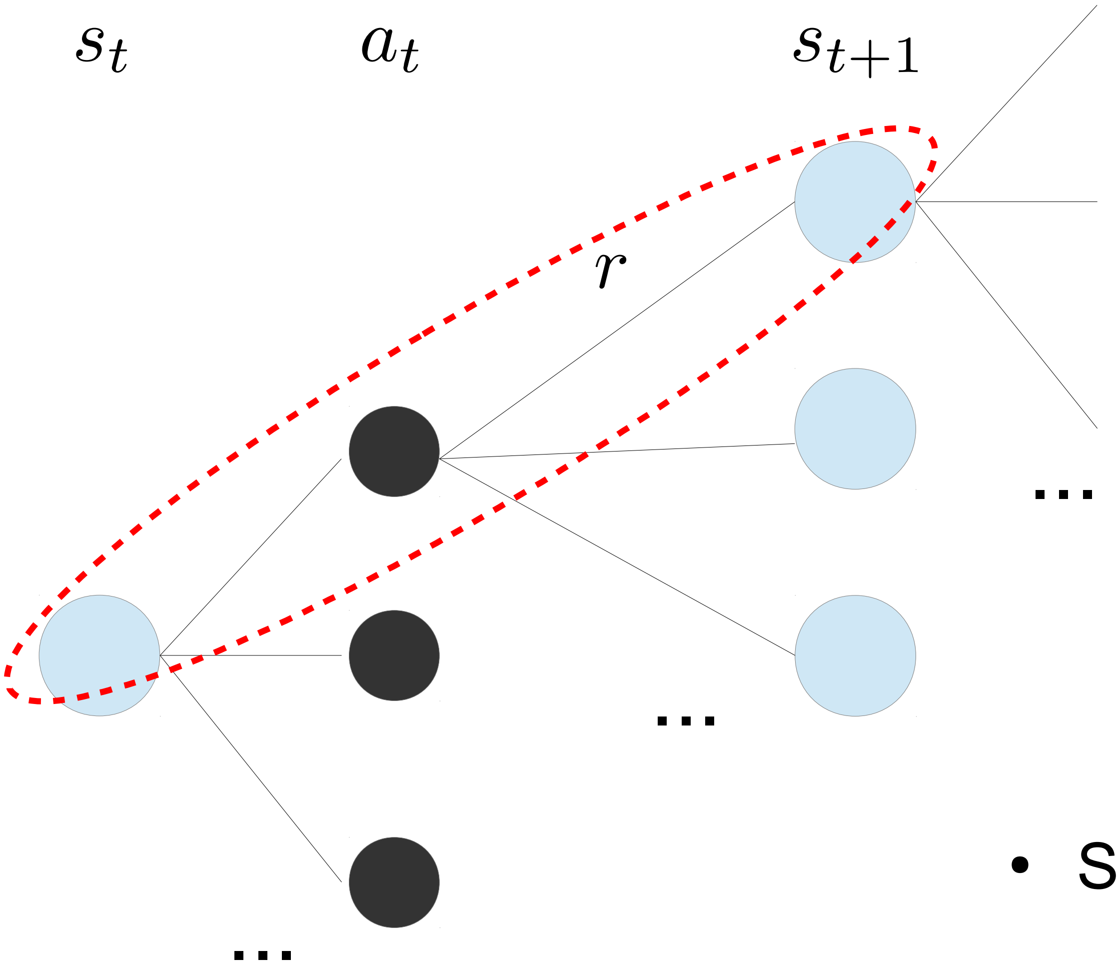
- State value function V

Backup Diagrams



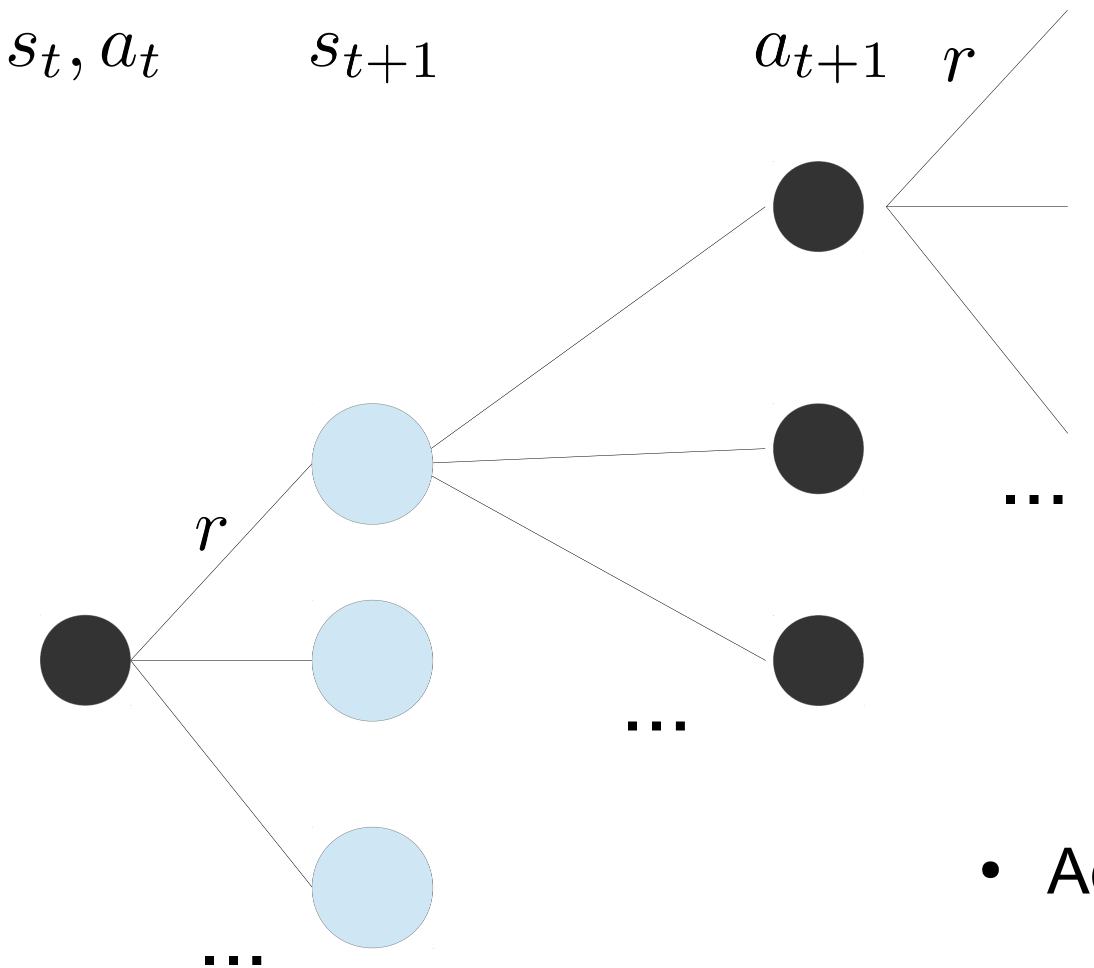
- State value function V

Backup Diagrams



- State value function V

Backup Diagrams



- Action value function Q

Policy Improvement

- Consider a given policy π
  - … can we improve it by changing the action taken at a specific state s ?
  - … yes if $Q^\pi(s,a) > V^\pi(s)$

- *[Policy Improvement Theorem]* Generally, for deterministic policies π , π', if

$$Q^\pi(s,\pi'(s)) \geq V^\pi(s),\ \forall s \in S$$

then

$$V^{\pi'}(s) \geq V^\pi(s),\ \forall s \in S$$

Policy Improvement (continued)

- A policy improvement step would then be:

$$\begin{aligned}\pi'(s) &= argmax_a Q^\pi(s,a) \\ &= argmax_a \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma \, V^\pi(s') \right]\end{aligned}$$

- Of course, this does not evaluate the value function for the new policy π', but if we put Policy Improvement and Policy Evaluation together, we get...

Policy Iteration

1. initialise V and π_0 (arbitrarily)

2. perform Policy Evaluation

3. perform Policy Iteration

4. if the policy has changed go to 2.

Value Iteration

- … is like Policy Iteration but with only a single backup of each state in the Policy Evaluation step.
- This still converges to an optimal policy.
- Policy Evaluation and Policy Improvement can be joined into a single update:

$$
\begin{aligned}
V_{k+1}(s) &= max_a E\{r_{t+1} + \gamma\, V_k(s_{t+1}) | s_t = s, a_t = a\} \\
&= max_a \sum_{s'} P^a_{ss'} \left[R^a_{ss'} + \gamma V_k(s')\right]
\end{aligned}
$$

- Need only compute the policy in the end.

Reading +

- Chapter 4 (up till 4.4) of Sutton and Barto (1st Edition) http://incompleteideas.net/book/ebook/the-book.html

- Please join Piazza for announcements and support: https://piazza.com/ed.ac.uk/spring2018/infr11010