Summary of Part I: Reinforcement Learning in Finite State and Action Spaces

Oliver Wallscheid

Common key ideas to all discussed rl methods so far

1. Estimating and comparing value functions
2. Backing up values along actual or possible state trajectories
3. Usage of GPI mechanism to maintain an approximate value function and policy trying to improve each of them on the basis of the other

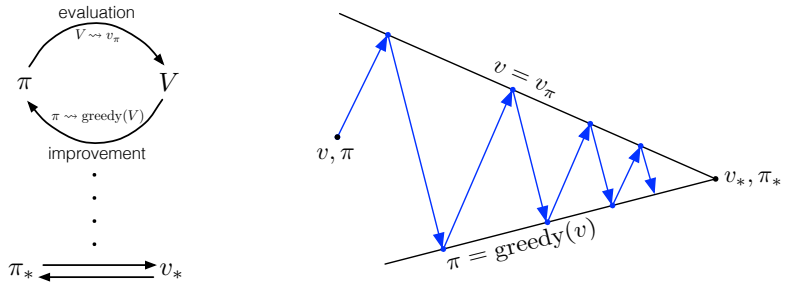

Fig. S-I.1: Generalized policy iteration (GPI) as a mutual building block of all previously discussed RL methods (source: R. Sutton and G. Barto, Reinforcement learning: an introduction, 2018, CC

two important rl dimensions: update depth and width

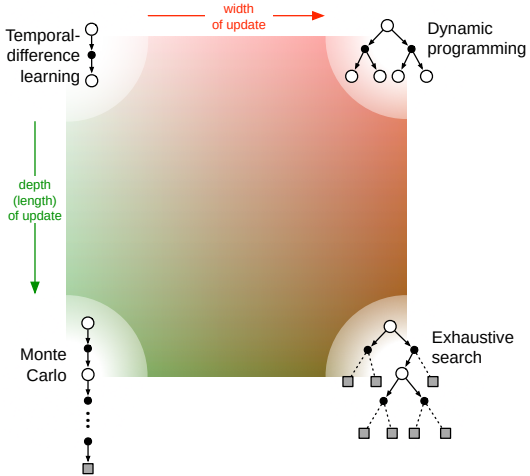

Fig. S-I.2: A slice through the RL method space (source: R. Sutton and G. Barto, Reinforcement learning: an introduction, 2018, CC BY-NC-ND 2.0)

Other important rl dimensions

Selected, non-exhaustive list:

- Problem space: How many states and actions? Stochastic vs. deterministic environment? Stationary?
- Policy objective: on-policy vs. off-policy? Explicit vs. implicit policy?
- Task: Episodic vs. continuing?
- Return definition: Discounting? General reward design?
- Value: State vs. action value estimation?
- Model: Required? Distribution vs. sample models? Learning vs. a priori (expert) knowledge?
- Exploration: How to search for new policies?
- Update order: synchronous vs. asynchronous? If latter, which order?
- Experience: simulated vs. real experience? Memory length and style?
- ...

Outlook

First part of the course:

Reinforcement learning on small finite action and state spaces

The problem space is such small that RL methods based on look-up tables are applicable.

Second part of the course::

Reinforcement learning using function approximators

The problem space is either continuous or contains an unfeasible large amount of discrete state-action pairs. Value estimates, models or explicit policies stored in look-up tables would let the memory demand explode. Modifications and extensions of available RL algorithms using function approximators are required.