Combine and conquer

DP and MC techniques both have desirable properties, that conflict with each other:

- Bootstrapping: Dynamic Programming updates value estimates based on other value estimates – efficient use of data, thanks to the optimal sub-problem structure. Note, in the case of RL here we look at episodic tasks and so we know that at the end of the episode we will be grounded by the actual return outcome.
- Sampling: Monte-Carlo methods sample rewards from the environment – no need to know the MDP.

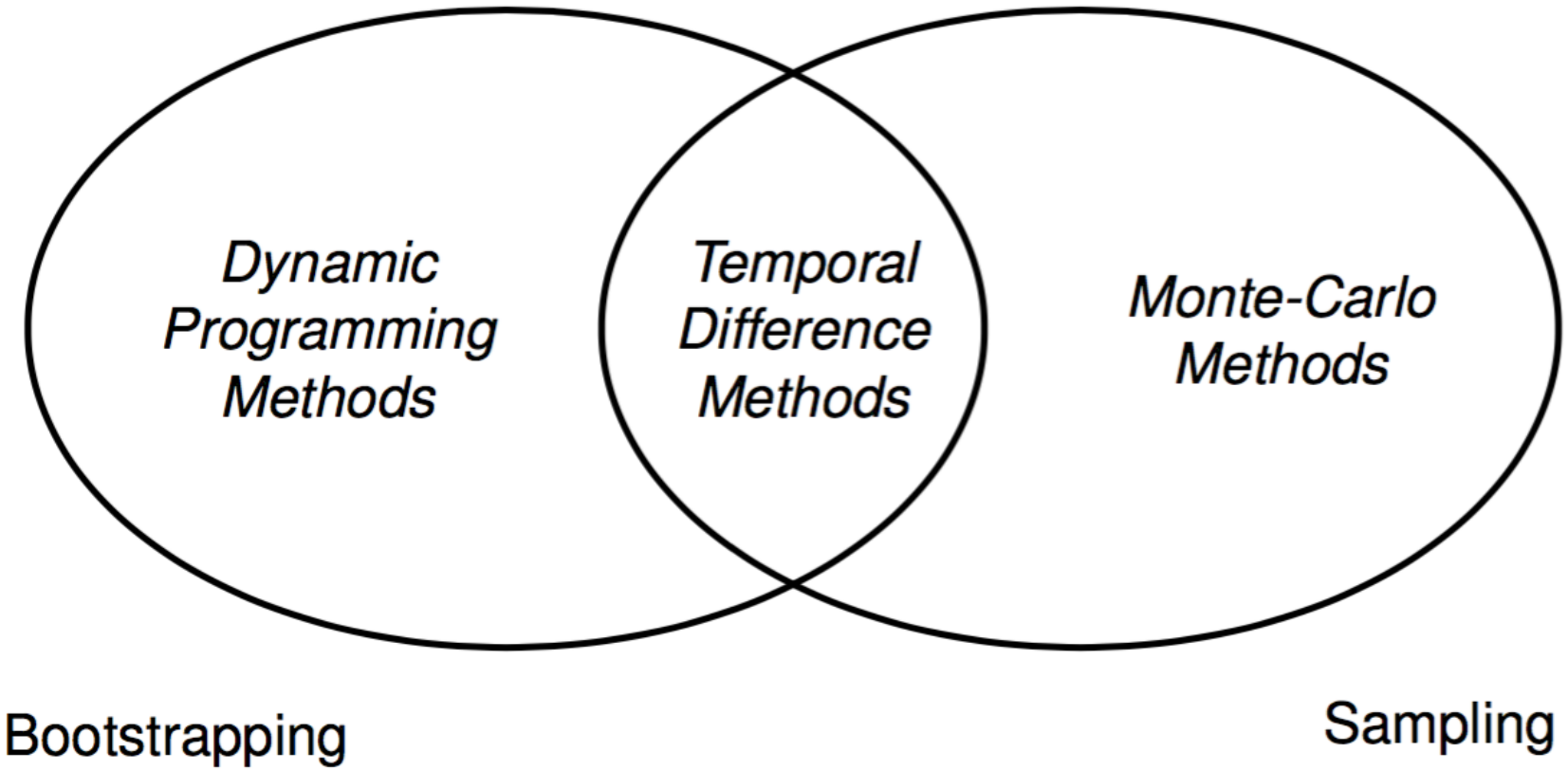
Can we have the cake and eat it?

Dynamic Programming performs full backups

TD performs bootstrapping from 1-step samples

Temporal Difference Learning

Temporal difference (TD) learning sits between Dynamic Programming and Monte-Carlo methods.



Temporal-Difference (TD) Learning

1. TD methods learn directly from episodes of experience (but also works for non-episodic tasks)
2. TD is model-free: no knowledge of MDP transitions or rewards needed
3. TD learns from incomplete episodes, by bootstrapping
4. TD updates a guess towards a guess

Temporal Difference Learning update rule

Recall that we use the following Monte-Carlo update

$$V(s_t) \leftarrow V(s_t) + \alpha(R_t - V(s_t)) \tag{36}$$

We update the value of $V(s_t)$ towards the actual return R_t. Note, how we use only measurements to form our estimates.
Temporal Difference methods perform a similar update after every time-step, i.e.

$$V(s_t) \leftarrow V(s_t) + \alpha\left(r_{t+1} + \gamma V(s_{t+1}) - V(s_t)\right) \tag{37}$$

We update the value of $V(s_t)$ towards the estimated return $r_{t+1} + \gamma V(s_{t+1})$ Note, how we are combining a measurements r_{t+1} with an estimate $V(s_{t+1})$ to produce a better estimate $V(s_t)$.

TDerminology

- $r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$ is the Temporal Difference Error.
- $r_{t+1} + \gamma V(s_{t+1})$ is the Temporal Difference Target

TD value function estimation Algorithm

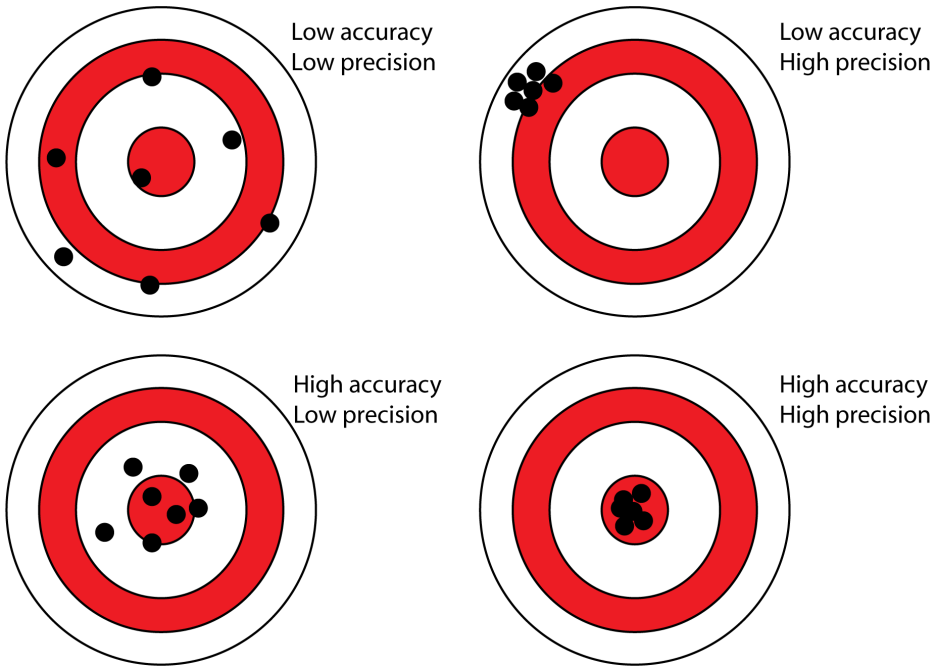
procedure TD-ESTIMATION(π)
  Init
    $\hat{V}(s) \leftarrow$ arbitrary value, for all $s \in S$.
  EndInit
  repeat(For each episode)
    Initialise s
    repeat(For each step of episode)
      a action chosen from π at s
      Take action a; observe r, and next state, s'
      $\delta \leftarrow r + \gamma \hat{V}(s') - \hat{V}(s)$
      $\hat{V}(s) \leftarrow \hat{V}(s) + \alpha\delta$
      $s \leftarrow s'$
    until s is absorbing state
  until Done
end procedure

Advantages & Disadvantages of MC & TD

1. TD can learn before knowing the final outcome ("you can back-up near-death")
   - TD can learn online after every step
   - MC must wait until end of episode before return is known
2. TD can learn without the final outcome
3. TD can learn from incomplete sequences
   - MC can only learn from complete sequences
   - TD works in continuing (non-terminating) environments MC only works for episodic (terminating) environments

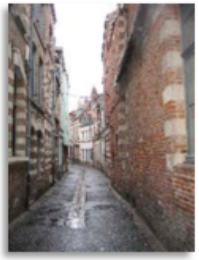
The Bias-Variance Tradeoff

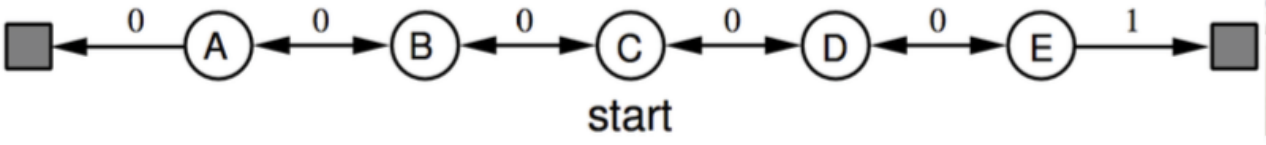
1. Sample return $R_t = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{T?1} r_T$ is unbiased estimate of $V^\pi(s_t)$
2. "True" TD target $r_{t+1} + \gamma V^\pi(s_{t+1})$ is unbiased estimate of $V^\pi(s_t)$
3. Estimated TD target $r_{t+1} + \gamma V(s_{t+1})$ is biased estimate of $V^\pi(s_t)$
4. TD target is much lower variance than the return:
   - Sample return depends on many random actions, transitions, rewards.
   - TD target depends on one random action, transition, reward



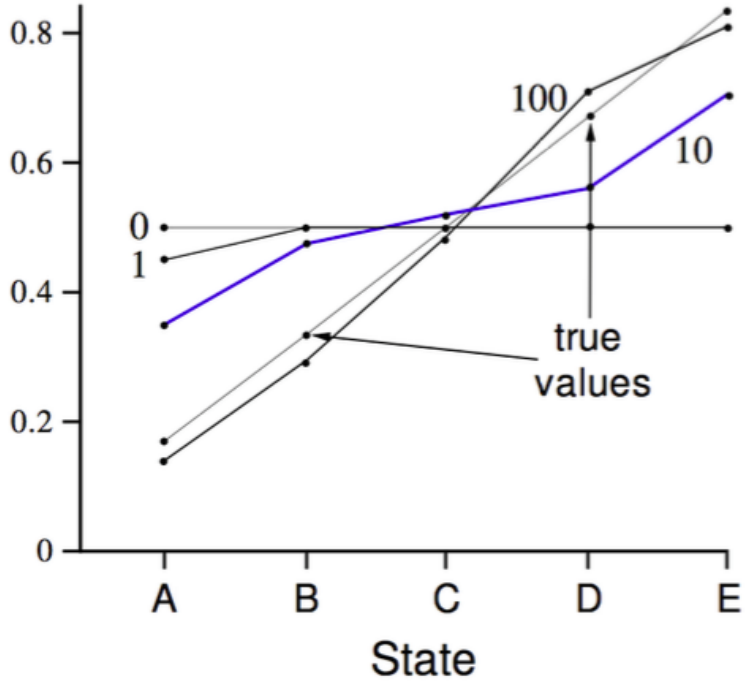
- MC has high variance, zero bias
  - Good convergence properties
  - even with function approximation (to be discussed later)
  - not very sensitive to initial value
  - very simple to understand and use
- TD exploits Markov property
  $\Rightarrow$ Usually more efficient in Markov environments
- TD has low variance, some bias
  - Usually more efficient than MC
  - TD – to be precise TD(0) – converges to $V^{\pi}(s)$
  - convergence not guaranteed with function approximation
  - more sensitive to initial value
- MC does not exploit Markov property
  $\Rightarrow$ Usually more effective in non-Markov environments

Random walk example: TD vs MC



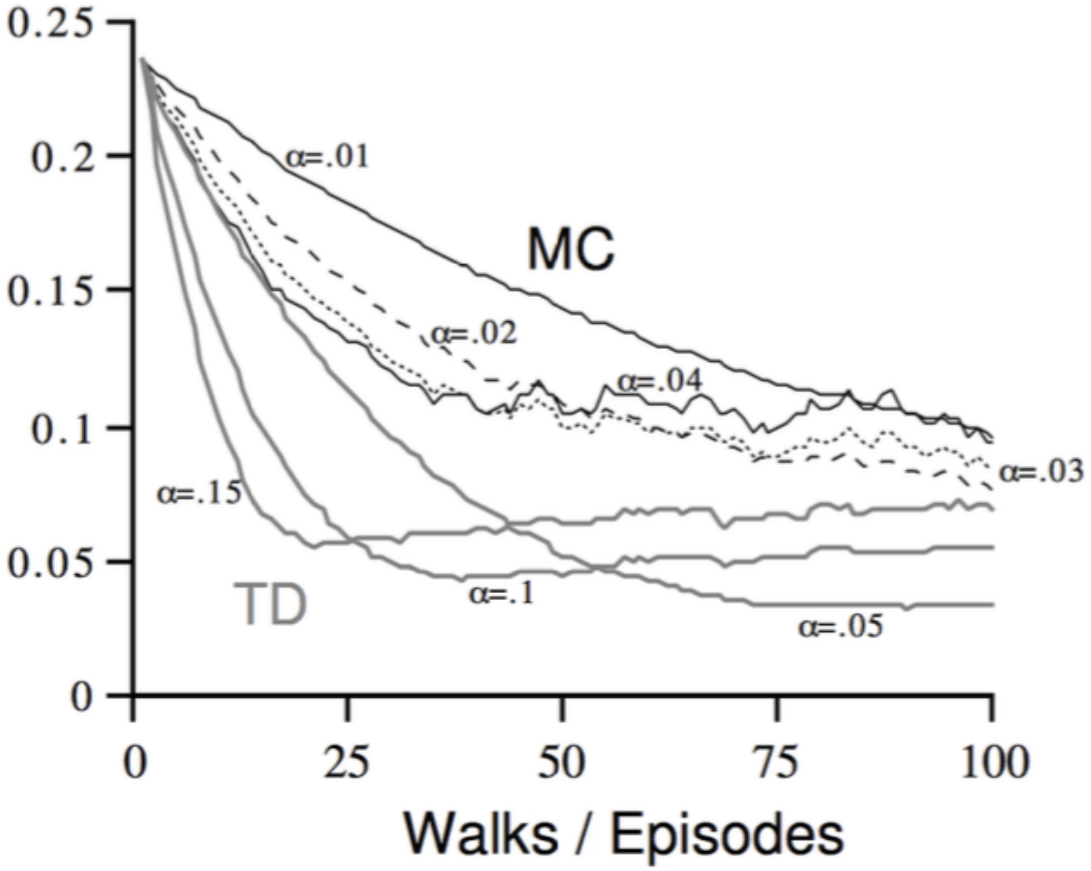


Values learned by TD(0) after various numbers of episodes



Random walk example: TD vs MC

RMS error, averaged over states



Data averaged over
100 sequences of episodes

Dynamic Programming performs full backups

TD performs bootstrapping from 1-step samples

Summary DP vs TD vs MC

Bootstrapping: update involves an *estimate*

- MC does not bootstrap
- DP bootstraps
- TD bootstraps

Sampling: update does not involve an *expected value*

- MC samples
- DP does not sample
- TD samples