OPTION-VALUE LEARNING ALGORITHMS

Jonathan Campbell

COMP-767

March 10, 2017

OVERVIEW

- Comparison of algorithms to estimate option-values.
  - SMDP Q-Learning
  - Intra-option Q-Learning
  - Combined SMDP/intra-option Q-Learning
  - Combined SMDP/intra-option SARSA

OPTIONS

- Temporally-abstract actions.
- An option is a triple:
  - states from which the option can be selected
  - policy to follow after the option is selected
  - probability of terminating in each state

SMDP Q-LEARNING

- Execute option o until termination.
  - Keep track of discounted reward *r* and number of timesteps *k*.
  - When option finished, update q-val for option:
    - ***s***: state in which option was chosen; ***s'***: state in which option finished

$$Q(s,o) \leftarrow Q(s,o) + \alpha\left[r + \gamma^k \max_{o' \in \mathcal{O}_{s'}} Q(s',o') - Q(s,o)\right]$$

  - Propagates value back to option start state.

INTRA-OPTION Q-LEARNING

- After each primitive action taken (during an option):
  - Update all options ***o*** that could have taken that action.

$$Q(s_t, o) \leftarrow Q(s_t, o) + \alpha\Big[\big(r_{t+1} + \gamma U(s_{t+1}, o)\big) - Q(s_t, o)\Big]$$

where

$$U(s, o) = \big(1 - \beta(s)\big) Q(s, o) + \beta(s) \max_{o' \in \mathcal{O}} Q(s, o').$$

- More efficient use of trajectories (more updates).

COMBINED SMDP/INTRA-OPTION Q-LEARNING

- Combine two update rules of SMDP and Intra-option.
  - When option terminates: use SMDP Q update rule.
  - At all times: use intra-option Q update rule.

- From Stolle's thesis (2004).

COMBINED SMDP/INTRA-OPTION SARSA

- Same as previous but:
  - Fix choice of next option o_{t+1}.
  - Use SARSA update rules:
    - SMDP:

$$Q(s,o) \leftarrow Q(s,o) + \alpha[r + \gamma^k Q(s', o_{t+1}) - Q(s,o)]$$

    - Intra-option:

$$Q(s_t,o) \leftarrow Q(s_t,o) + \alpha[(r_{t+1} + \gamma U(s_{t+1},o)) - Q(s_t,o)]$$

$$\text{where } U(s,o) = (1-\beta(s))\,Q(s,o) + \beta(s)\,Q(s,o_{t+1})$$

IMPLEMENTATION

- Uses Gridworld RL framework from UC Berkeley AI course
    - http://ai.berkeley.edu/reinforcement.html
- Extension to options.

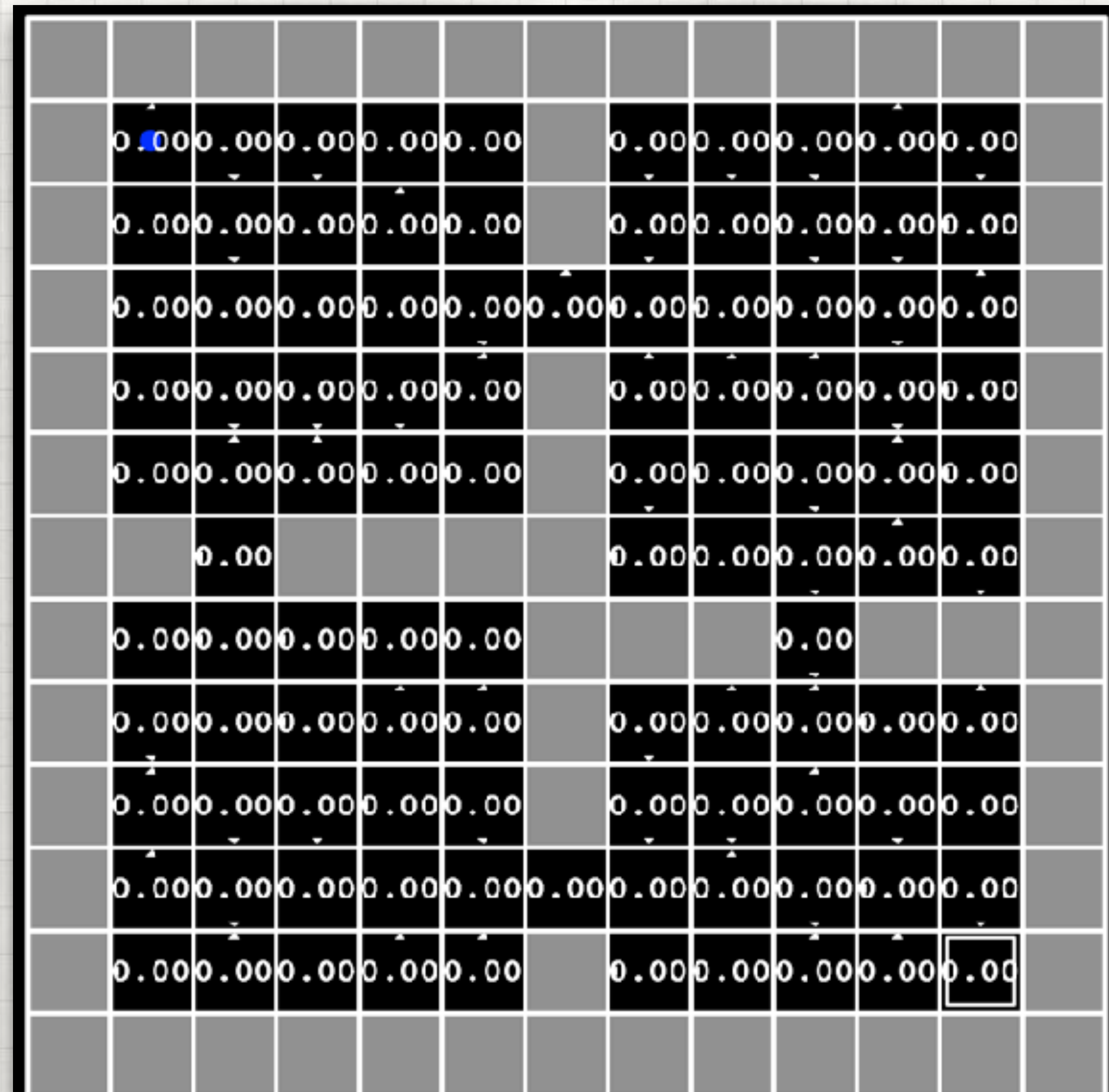
METHOD

- Gridworld "rooms example" used for experiments.
  - 4 rooms connected by single-tile hallway.
  - Goal in one room.
  - 4 single-movement primitive options (available everywhere).
  - 4 "hallway options" per room:
    - Two per hallway: takes shortest distance to that hallway.
    - One prefers horizontal movement, one prefers vertical.

ROOMS EXAMPLE

```
grid = [['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
        ['#', 'S', ' ', ' ', ' ', ' ', '#', ' ', ' ', ' ', ' ', ' ', '#'],
        ['#', ' ', ' ', ' ', ' ', ' ', '#', ' ', ' ', ' ', ' ', ' ', '#'],
        ['#', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '#'],
        ['#', ' ', ' ', ' ', ' ', ' ', '#', ' ', ' ', ' ', ' ', ' ', '#'],
        ['#', ' ', ' ', ' ', ' ', ' ', '#', ' ', ' ', ' ', ' ', ' ', '#'],
        ['#', '#', ' ', '#', '#', '#', '#', ' ', ' ', ' ', ' ', ' ', '#'],
        ['#', ' ', ' ', ' ', ' ', ' ', '#', '#', '#', ' ', '#', '#', '#'],
        ['#', ' ', ' ', ' ', ' ', ' ', '#', ' ', ' ', ' ', ' ', ' ', '#'],
        ['#', ' ', ' ', ' ', ' ', ' ', '#', ' ', ' ', ' ', ' ', ' ', '#'],
        ['#', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '#'],
        ['#', ' ', ' ', ' ', ' ', ' ', '#', ' ', ' ', ' ', ' ', +1,  '#'],
        ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#']]
```

ROOMS EXAMPLE

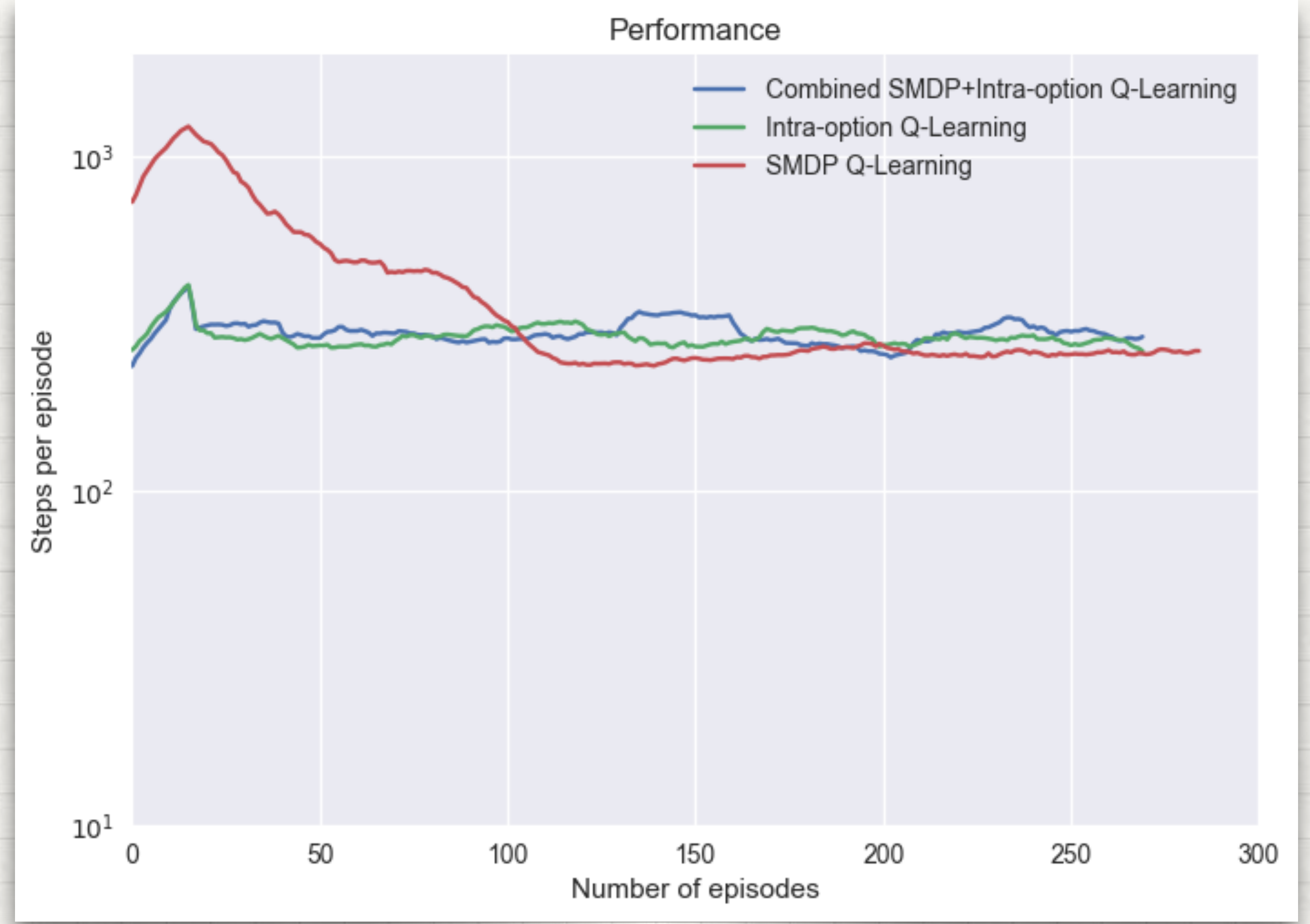


METHOD

```
Option(policyFn=lambda state: 'north',
       terminationFn=lambda state: True,
       initiationSet=all_states,
       primitive=True, primitiveName='north'),
```

```
Option(policyFn=lambda state: getMovementDelta(state, west_corridor, pathfind_grid),
       terminationFn=lambda state: not state in getRoomStates(1),
       initiationSet=getRoomStates(1) + [north_corridor]),
```

TIME COMPARISON



COMPARISON ANALYSIS

- SMDP takes longer to converge to optimal.
  - Low % of primitive options contributes to this difference.
    - (SMDP == Intra-option when all opts. primitive)
- No difference between intra-option and combined:
  - Environment may be too simple to show meaningful difference.

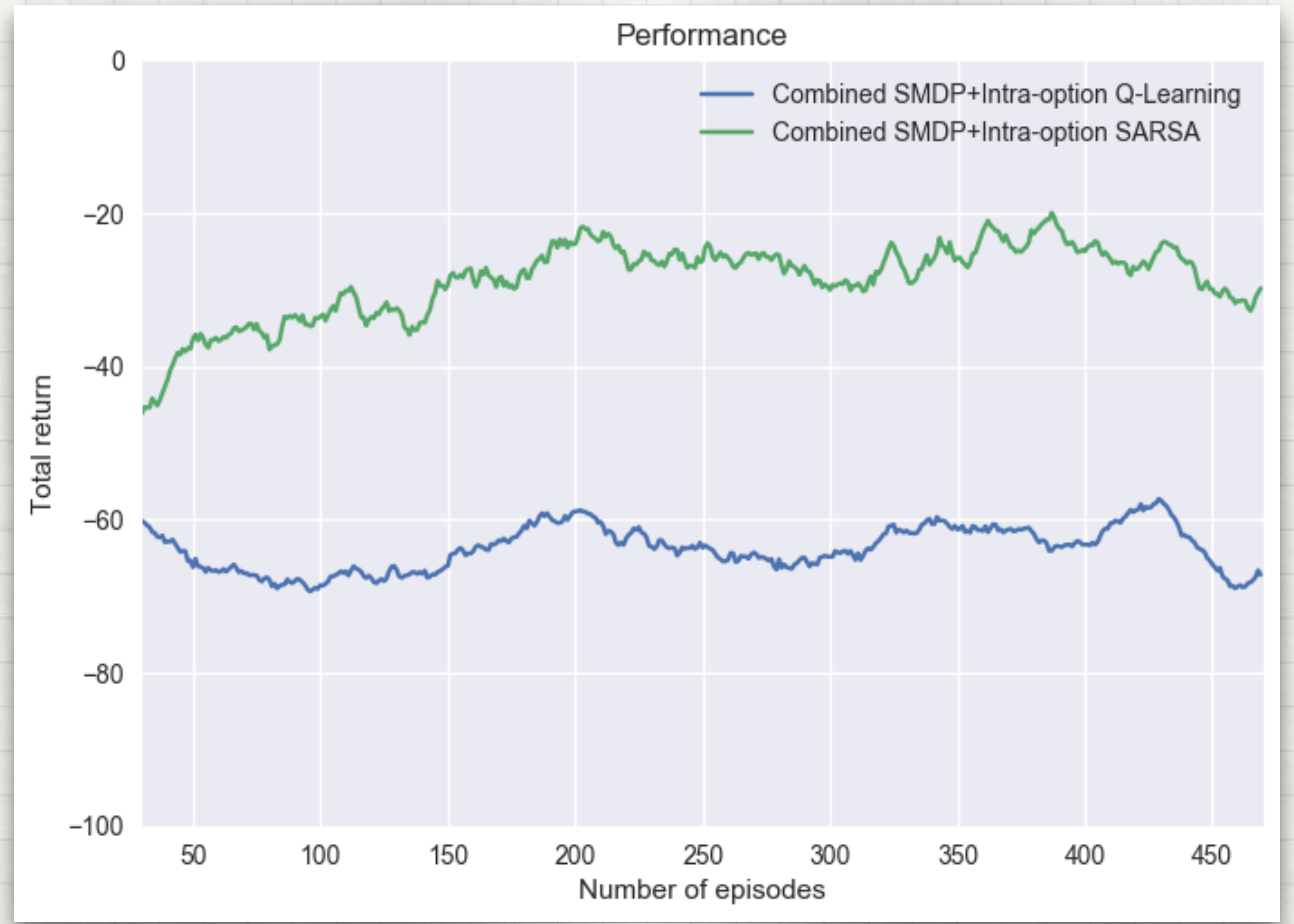
Q-LEARNING VS SARSA

- Same as earlier rooms example, except:
  - Noise in actions: 1/3 chance to move in wrong direction.
  - Middle corners of rooms are terminal states (negative reward).
  - Goal in right-most hallway.

SCARY ROOMS EXAMPLE

```
grid = [['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#'],
        ['#', 'S', ' ', ' ', ' ', ' ', '#', ' ', ' ', ' ', ' ', ' ', '#'],
        ['#', ' ', ' ', ' ', ' ', ' ',-100, ' ', ' ',-100,-100, ' ', '#'],
        ['#', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',-100, ' ', '#'],
        ['#', ' ', ' ', ' ', ' ', ' ',-100, ' ', ' ', ' ', ' ', ' ', '#'],
        ['#', ' ', ' ', ' ', ' ', ' ',-100, ' ', ' ', ' ', ' ', ' ', '#'],
        ['#', '#', ' ', '#',-100,-100,-100, ' ', ' ', ' ', ' ', ' ', '#'],
        ['#', ' ', ' ', ' ', ' ', ' ',-100,-100,-100, +10,-100, '#', '#'],
        ['#', ' ', ' ', ' ', ' ', ' ',-100, ' ', ' ', ' ', ' ', ' ', '#'],
        ['#', ' ', ' ', ' ', ' ', ' ', '#', ' ', ' ', ' ', ' ', ' ', '#'],
        ['#', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '#'],
        ['#', ' ', ' ', ' ', ' ', ' ', '#', ' ', ' ', ' ', ' ', ' ', '#'],
        ['#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#', '#']]
```

Q-LEARNING VS SARSA: AVG. TOTAL RETURN



CODE

- Available at course GitHub repo:

  https://github.com/rllabmcgill/rlcourse-march-10-campbelljc