HW4

March 1, 2021

Max Schrader

03/01/2020

[1]:
```
states = [j for j in range(16)]
v_k_master = [0. for _ in range(16)]
terminal_states = (0, 15)
# rewards = [0 if i in terminal_states else 0 for i in range(16)]
action_reward = -1
discount_factor = 1
possible_actions = [-1, 1, -4, 4]
possible_action_master = [possible_actions.copy() if i not in terminal_states␣
 ↪else [] for i in range(16)]
best_actions_master = [possible_actions.copy() if i not in terminal_states else␣
 ↪[] for i in range(16)]
action_probabilty = 0.25
```

[2]:
```
def pretty_print_table(table):
    for i in range(16):
        if (i + 4) % 4 == 0:
            print([round(val, 1) for val in table[i:i+4]])
```

[3]:
```
def get_potential_actions(current_state, action_list):
    # this is terible programming. Should be a way to do this without all these␣
 ↪if else statements
    if action_list:
        for i, action in enumerate(action_list):
            if (current_state == 1) and (action == -1):
                yield current_state - 1
            elif (current_state % 4 == 0) and (action == -4):
                yield current_state - 4
            elif (current_state % 4 == 0) and (action == -1):
                yield current_state
            elif (current_state + action) % 4 == 0:
                # this is a wrap around, not allowed!
                yield current_state
            elif ((current_state in [0, 1, 2, 3]) and (action == -4)) or \
            ((current_state in [12, 13, 14, 15]) and (action == 4)):
                yield current_state
            else:
                yield current_state + action
```

[4]:
```
def greedy(k, action_reward, v_k, terminal_states, actions, action_probabilty,␣
 ↪best_actions):
    for _ in range(k):
        v_k_last = v_k.copy()
        for i, (current_reward, action_space) in enumerate(zip(v_k, actions)):
            if i not in terminal_states:
                policy_reward = action_probabilty * len(action_space) *␣
 ↪action_reward
                reward = [action_probabilty * v_k_last[new_state] for new_state␣
 ↪in get_potential_actions(i, action_space)]
#                 print(i, list(get_potential_actions(i, action_space)), reward)
                v_k[i] = sum(reward) + policy_reward
    return v_k

```

0.1 K = 0

[5]:
```
k = 0
v_k = greedy(k, action_reward, v_k_master.copy(), terminal_states,␣
 ↪possible_action_master.copy(), action_probabilty, best_actions_master.copy())
pretty_print_table(v_k)
```

```
[0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0]
```

0.2 K = 1

[6]:
```
k = 1
v_k = greedy(k, action_reward, v_k_master.copy(), terminal_states,␣
 ↪possible_action_master.copy(), action_probabilty, best_actions_master.copy())
pretty_print_table(v_k)
```

```
[0.0, -1.0, -1.0, -1.0]
[-1.0, -1.0, -1.0, -1.0]
[-1.0, -1.0, -1.0, -1.0]
[-1.0, -1.0, -1.0, 0.0]
```

0.3 K = 2

[7]:
```
k = 2
v_k = greedy(k, action_reward, v_k_master.copy(), terminal_states,␣
 ↪possible_action_master.copy(), action_probabilty, best_actions_master.copy())
pretty_print_table(v_k)
```

```
[0.0, -1.8, -2.0, -2.0]
[-1.8, -2.0, -2.0, -2.0]
[-2.0, -2.0, -2.0, -1.8]
[-2.0, -2.0, -1.8, 0.0]
```

0.4 K = 3

[8]:
```
k = 3
v_k = greedy(k, action_reward, v_k_master.copy(), terminal_states,␣
 ↪possible_action_master.copy(), action_probabilty, best_actions_master.copy())
pretty_print_table(v_k)
```

```
[0.0, -2.4, -2.9, -3.0]
[-2.4, -2.9, -3.0, -2.9]
[-2.9, -3.0, -2.9, -2.4]
[-3.0, -2.9, -2.4, 0.0]
```

0.5 K = 5

[9]:
```
k = 5
v_k = greedy(k, action_reward, v_k_master.copy(), terminal_states,␣
 ↪possible_action_master.copy(), action_probabilty, best_actions_master.copy())
pretty_print_table(v_k)
```

```
[0.0, -3.7, -4.7, -4.9]
[-3.4, -4.7, -4.8, -4.7]
[-4.6, -4.8, -4.5, -3.7]
[-4.9, -4.7, -3.7, 0.0]
```

0.6 K = 10

[10]:
```
k = 10
v_k = greedy(k, action_reward, v_k_master.copy(), terminal_states,␣
 ↪possible_action_master.copy(), action_probabilty, best_actions_master.copy())
pretty_print_table(v_k)
```

```
[0.0, -6.3, -8.5, -9.0]
[-5.5, -8.4, -8.6, -8.4]
[-7.9, -8.6, -7.8, -6.2]
[-8.8, -8.2, -6.1, 0.0]
```

0.7 K = 20

[11]:
```
k = 20
v_k = greedy(k, action_reward, v_k_master.copy(), terminal_states,␣
 ↪possible_action_master.copy(), action_probabilty, best_actions_master.copy())
pretty_print_table(v_k)
```

```
[0.0, -10.0, -13.7, -14.7]
[-8.2, -13.3, -13.8, -13.5]
[-12.3, -13.6, -12.3, -9.6]
[-13.8, -12.8, -9.4, 0.0]
```

0.8 K = ∞

[12]:
```
k = 100000
v_k = greedy(k, action_reward, v_k_master.copy(), terminal_states,␣
 ↪possible_action_master.copy(), action_probabilty, best_actions_master.copy())
pretty_print_table(v_k)
```

```
[0.0, -15.3, -21.3, -23.2]
[-12.3, -20.6, -21.4, -21.1]
[-18.6, -21.0, -18.8, -14.6]
[-21.1, -19.5, -14.1, 0.0]
```