Recitation 3: Homework 1

Yafei Hu and Justin Kiefel

Problem 1: Value Iteration & Policy Iteration

1.1: Contraction Mapping

An operator F on a normed vector space $\mathcal{X}$ is a γ-contraction, for $0 < \gamma < 1$ provided for all $x, y \in \mathcal{X}$:

$$\|F(x) - F(y)\| \leq \gamma \|x - y\|$$

Theorem (Contraction mapping)
For a γ-contraction F in a complete normed vector space $\mathcal{X}$:

- F converges to a unique fixed point in $\mathcal{X}$,
- at a linear convergence rate γ.

Problem 1: Value Iteration & Policy Iteration

Fixed point definition:

A point/vector $x \in \mathcal{X}$ is a fixed point of an operator F if $F(x) = x$

Banach Fixed Point Theorem:

If F is a γ-contraction mapping, then:

- F has a **unique** fixed point
- $\forall x_0 \in \mathcal{X}$, the sequence $x_{n+1} = F(x_n)$ converges to x^* in a geometric fashion:

$$||x_n - x^*|| \leq \gamma^n ||x_0 - x^*||$$

thus $\lim_{n \to \infty} ||x_n - x^*|| \leq \lim_{n \to \infty} (\gamma^n ||x_0 - x^*||) = 0$

Ref: Lecture 4 of DeepMind & UCL RL Lecture Series 2021 https://deepmind.com/learning-resources/reinforcement-learning-series-2021

Problem 1: Value Iteration & Policy Iteration

Policy Evaluation

Iterative Policy Evaluation, for estimating $V \approx v_\pi$

Input π, the policy to be evaluated
Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
    $\Delta \leftarrow 0$
    Loop for each $s \in \mathcal{S}$:
        $v \leftarrow V(s)$
        $V(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s', r | s, a) \big[r + \gamma V(s')\big]$
        $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Problem 1: Value Iteration & Policy Iteration

Policy Iteration

Note the difference between sync and async Policy Iteration

Policy Iteration (using iterative policy evaluation) for estimating $\pi \approx \pi_*$

1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Loop:
   $\quad \Delta \leftarrow 0$
   $\quad$ Loop for each $s \in \mathcal{S}$:
   $\qquad v \leftarrow V(s)$
   $\qquad V(s) \leftarrow \sum_{s',r} p(s',r|s,\pi(s))\big[r + \gamma V(s')\big]$
   $\qquad \Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
   $\quad$ *old-action* $\leftarrow \pi(s)$
   $\quad \pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s',r|s,a)\big[r + \gamma V(s')\big]$
   $\quad$ If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

Problem 1: Value Iteration & Policy Iteration

Value Iteration

Note the difference between sync and async Value Iteration

Value Iteration, for estimating $\pi \approx \pi_*$

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
| $\Delta \leftarrow 0$
| Loop for each $s \in \mathcal{S}$:
| $v \leftarrow V(s)$
| $V(s) \leftarrow \max_a \sum_{s',r} p(s', r | s, a)\left[r + \gamma V(s')\right]$
| $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Output a deterministic policy, $\pi \approx \pi_*$, such that
  $\pi(s) = \operatorname{argmax}_a \sum_{s',r} p(s', r | s, a)\left[r + \gamma V(s')\right]$

Problem 1: Value Iteration & Policy Iteration

Synchronous and Asynchronous Policy Iteration/Value Iteration

- Synchronous value iteration stores two copies of value function
  - for all s in $\mathcal{S}$

$$v_{new}(s) \leftarrow \max_{a \in \mathcal{A}} \left(r(s,a) + \gamma \sum_{s' \in \mathcal{D}} p\left(s' \mid s, a\right) v_{old}\left(s'\right) \right)$$

$$v_{old} \leftarrow v_{new}$$

- In-place value iteration only stores one copy of value function
  - for all s in $\mathcal{S}$

$$v(s) \leftarrow \max_{a \in \mathcal{A}} \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} T\left(s' \mid s, a\right) v\left(s'\right) \right)$$

Problem 1: Value Iteration & Policy Iteration

Synchronous and Asynchronous Policy Iteration/Value Iteration

Example: synchronous and randperm asynchronous Value Iteration in 4 × 4 Frozen Lake Environment

Synchronous

Iteration 1

Previous value function

0.	0.	0.	0.
0.	0.	0.	0.
0.	0.	0.	0.
0.	0.	0.	0.

Updated value function

0.	0.	0.	0.
1.	0.	0.	0.
0.	0.	0.	0.
0.	0.	0.	0.

Asynchronous

Updated value function

0.9	0.	0.	0.
1.	0.	0.	0.
0.9	0.	0.	0.
0.81	0.	0.	0.

Iteration 2

0.	0.	0.	0.
1.	0.	0.	0.
0.	0.	0.	0.
0.	0.	0.	0.

0.9	0.	0.	0.
1.	0.	0.	0.
0.9	0.	0.	0.
0.	0.	0.	0.

0.9	0.	0.	0.
1.	0.	0.	0.
0.9	0.	0.	0.
0.81	0.729	0.	0.

Iteration 3

0.9	0.	0.	0.
1.	0.	0.	0.
0.9	0.	0.	0.
0.	0.	0.	0.

0.9	0.	0.	0.
1.	0.	0.	0.
0.9	0.	0.	0.
0.81	0.	0.	0.

0.9	0.	0.	0.
1.	0.	0.	0.
0.9	0.	0.	0.
0.81	0.729	0.6561	0.

Problem 1: Value Iteration & Policy Iteration

Problem 1.5: Manhattan distance as heuristic function

1. Compute the heuristic function for all the states in advance
2. Sweep the states ordered by the pre-computed heuristic function (in this case, Manhattan distance)

   Manhattan distance between some state *S* and goal *G*

$$d(S,G) = \sum_{i=1}^{n} |S_i - G_i|, \text{ where } n = 2$$

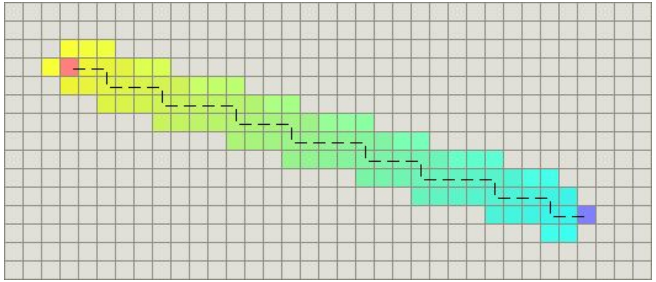

Illustration of Manhattan distance as heuristic [1]

More about heuristic: [1] http://theory.stanford.edu/~amitp/GameProgramming/Heuristics.html

Problem 1 Q & A

1. Will I get the same iteration number in PI/VI and same policy every time?

   Yes, you are expected to get same results every time you run the experiment.

Problem 2: Bandits

Estimating Expected Reward

$$\mathbb{E}\{R_t\} = \frac{1}{20}\Sigma_{k=1}^{20} R_t^k$$

- Average of rewards received at a given time step
- Unbiased
- High Variance

$$\mathbb{E}\{R_t\} = \frac{1}{20}\Sigma_{k=1}^{20} \mathbb{E}\{r^k(A_t^k)|\pi_t^k\}$$

- Average of expected rewards conditioned on the policy
- Unbiased
- Lower Variance
- Remember to still use R_t for the agent’s update

Efficient Q-Updates

$$
\begin{aligned}
Q_{n+1} &= \frac{1}{n}\sum_{i=1}^{n} R_i \quad \text{Don't Use This} \\
&= \frac{1}{n}\left(R_n + \sum_{i=1}^{n-1} R_i\right) \\
&= \frac{1}{n}\left(R_n + (n-1)\frac{1}{n-1}\sum_{i=1}^{n-1} R_i\right) \\
&= \frac{1}{n}\Big(R_n + (n-1)Q_n\Big) \\
&= \frac{1}{n}\Big(R_n + nQ_n - Q_n\Big) \\
\text{Use This} \quad &= Q_n + \frac{1}{n}\Big[R_n - Q_n\Big],
\end{aligned}
$$

Problem 2.7 - Correlated Rewards

I.I.D. Rewards

$$r(k) \sim \mathcal{N}(\mu, \sigma^2) \; \forall k \in [K]$$

Correlated Rewards

$$[r(1) \dots r(K)]^T \sim \mathcal{N}(\mu_0, \Sigma_0)$$

Non-Diagonal

Questions?